

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**Desenvolvimento de Algoritmos e
Ferramentas de Síntese e Validação de
Estratégias para o Controle em Manobras
ou Navegação de Veículos ou Robôs Móveis
Multi-Articulados**

Tiago Reinan Barreto de Oliveira

Vitória

31 de Agosto de 2010

Tiago Reinan Barreto de Oliveira

**Desenvolvimento de Algoritmos e
Ferramentas de Síntese e Validação de
Estratégias para o Controle em Manobras
ou Navegação de Veículos ou Robôs Móveis
Multi-Articulados**

Universidade Federal do Espírito Santo
Centro Tecnológico
Programa de Pós-Graduação em Engenharia Elétrica

Vitória

31 de Agosto de 2010

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Oliveira, Tiago Reinan Barreto de, 1980-
O48d Desenvolvimento de algoritmos e ferramentas de síntese e
validação de estratégias para o controle em manobras ou
navegação de veículos ou robôs móveis multi-articulados / Tiago
Reinan Barreto de Oliveira. – 2010.
122 f. : il.

Orientador: Edson de Paula Ferreira.
Dissertação (mestrado) – Universidade Federal do Espírito
Santo, Centro Tecnológico.

1. Robôs móveis. 2. Navegação de robôs móveis. 3. Sistemas
de veículos auto-guiados. 4. Sistemas difusos. 5. Sistemas
inteligentes de controle. I. Ferreira, Edson de Paula. II.
Universidade Federal do Espírito Santo. Centro Tecnológico. III.
Título.

CDU: 621.3

Tiago Reinan Barreto de Oliveira

Desenvolvimento de Algoritmos e Ferramentas de Síntese e Validação de Estratégias para o Controle em Manobras ou Navegação de Veículos ou Robôs Móveis Multi-Articulados

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Engenharia Elétrica - Automação.

Aprovada em 31 de Agosto de 2010.

Comissão Examinadora:



Prof. Dr. Edson de Paula Ferreira
Universidade Federal do Espírito Santo, Orientador



Prof. Dr. Celso José Munaro
Universidade Federal do Espírito Santo



Prof. Dr. Hans Rolf Kulitz
Centro Federal de Educação Tecnológica do Espírito Santo

Vitória, 31 de Agosto de 2010.

Dedicatória

A Deus, autor e regente do universo. Aos meus pais, Cleber e Noélia, e minha irmã Karol, pelo carinho, incentivo e apoio, sem os quais dificilmente conseguiria êxito. A Marcela, pelo amor, paciência e por compartilhar parte da sua vida comigo. Ao meu orientador, Prof Dr. Edson, pelo apoio nos momentos difíceis e pelos conhecimentos científicos transmitidos, mas sobretudo, pela figura humana que é.

Agradecimentos

Meus agradecimentos sinceros:

À minha família, por tudo. Serei eternamente grato a vocês!

À minha namorada Marcela, pelo incentivo e pela ajuda que permitiu que eu me dedicasse mais tempo a essa dissertação. Muito obrigado.

Ao meu orientador, Prof. Dr. Edson, pela compreensão, confiança e orientação. Minha profunda gratidão Edinho.

Ao meu primeiro orientador, Prof. Dr. Ailson, pelo grande aprendizado adquirido, mesmo em pouco tempo de trabalho juntos. Abraço Prof.

Agradeço também à Mariana pela amizade, conselhos e empenho nas atividades de pesquisa. Ao Douglas pelo produtivo período, apesar de curto, em que trabalhamos.

Ao amigo e companheiro de LAI2, Filipe, pela amizade, profissionalismo e dedicação nos trabalhos. Abraço meu camarada!

Ao Edson Barbosa pela doação do automodelo e a FACITEC pelo apoio financeiro ao projeto.

A todos os professores que tive, que em sua grande maioria, eram pessoas comprometidas e competentes na arte de possibilitar o aprendizado.

Finalmente a todos os demais amigos que quase sempre entenderam a minha ausência e me incentivaram a concluir esse trabalho. Muito obrigado!

Tiago Reinan Barreto de Oliveira

“Muitos dos fracassos da vida ocorrem com as pessoas que não reconheceram o quão próximas elas estavam do sucesso quando desistiram”.

Thomas A. Edison

Lista de Tabelas

1	Principais características do Atmega32.	11
2	Especificações da placa de controle.	12
3	Características do módulo de comunicação.	32
4	Base de regras dos controladores.	43
5	Base de regras do controlador de α_3	47
6	Número de regras geradas por cada controlador em função do número de articulações e da quantidade de termos empregados no particionamento das variáveis de entrada.	49

Lista de Figuras

1	Exemplo do uso de veículo multi-Articulado no transporte de cargas. . . .	2
2	(a)Caminhão não articulado (b)Veículo biarticulado (c) Rodo Trem com quatro <i>trailers</i> (d) Rodo Trem com seis <i>trailers</i>	3
3	Automodelo <i>Globe Liner</i> , elemento trator.	9
4	Veículo multi-articulado montado	10
5	(a) <i>Layout</i> da placa de circuito (b) Placa devidamente montada.	12
6	Esquema de ligação entre a placa de controle e os periféricos.	13
7	(a) Quadratura (b) Absoluto com codificação <i>Gray</i> (c) Incremental. . . .	14
8	Cicuito gerador de pulsos do encoder	15
9	Local de instalação do <i>encoder</i> , destaque para o fototransistor e o amplificador operacional.	16
10	Controlador Eletrônico de Velocidade.	17
11	Diagrama de blocos do controlador de velocidade implementado.	18
12	Resposta do controlador de velocidade implementado.	18
13	Potenciômetro instalado na junta articulada.	19
14	Esquema de Triangulação utilizado pelo sensor infravermelho.	21
15	Esquema de Triangulação utilizado pelo sensor infravermelho.	22
16	(a) Anteparo de calibração (b) Curva de calibração para o sensor <i>GP2D12</i>	23
17	Sinal de controle do servomotor.	24
18	Aparato de calibração da direção.	25
19	(a) Resposta inicial do sistema de direção (b) Primeira correção (c) Segunda correção (d) Resposta final do sistema corrigido.	26

20	Fluxograma das principais rotinas do <i>firmware</i>	28
21	Arquitetura de processamento e comunicação.	30
22	(a) Transceptor TRF 2.4Ghz e (b)Layout da placa de circuito (c) Módulo de comunicação montado.	31
23	Datagrama de comunicação.	32
24	Placa USB conectado ao computador.	33
25	Definição dos ângulos e dimensões do protótipo. Fonte: Adaptado de [1].	36
26	Controlador CpPI.	37
27	Controlador Global com PID.	38
28	Formato das partições de todas as variáveis. Partições de α_1 em (a), α_2 em (b), α_3 em (c) e <i>Erro</i> em (d).	39
29	Arquitetura do CGPR inicialmente proposta.	42
30	Particionamento das variáveis. Em (a) <i>Erro</i> α_2 . Em (b) h_1 . Em (c) <i>Erro</i> α_3 . Em (d) h_2	44
31	Variação da velocidade de abertura e fechamento de α_2	45
32	Arquitetura do controlador CGPR para duas articulações.	46
33	Estrutura geral do CGPR.	46
34	Particionamento de α_2	46
35	Superfície de decisão para o controlador <i>fuzzy</i>	48
36	Ferramentas utilizadas nos experimentos	51
37	$Ref_{\alpha_2} = 0^\circ$ (a) Compensador proporcional (b)Compensador proporcional e integral	53
38	$Ref_{\alpha_2} = 25^\circ$ (a) Compensador Proporcional (b)Compensador Proporcional e Integral	54
39	Referência em rampa para α_2 (a) Compensador Proporcional (b)Compensador Proporcional e Integral	54
40	(a) Compensador proporcional (b)Compensador proporcional e integral (c) Compensador proporcional, Integral e derivativo	55

41	(a) Compensador proporcional (b)Compensador proporcional e integral (c) Compensador proporcional, Integral e derivativo	56
42	Arquitetura final do controlador CGPR.	56
43	<i>Ref</i> $\alpha_3 = 0^\circ$ (a) Controlador Global (b)Controlado CGPR	57
44	<i>Ref</i> $\alpha_3 = 0^\circ$ (a) Controlador Global - Engavetamento (b)Controlado CGPR (c) Controlador CGPR - Engavetamento	58
45	<i>Ref</i> $\alpha_3 = -20^\circ$ (a) Controlador Global (b)Controlado CGPR	59
46	<i>Ref</i> $\alpha_3 = -30^\circ$ (a) Controlador Global (b)Controlado CGPR	59
47	(a) Controlador Global - Engavetamento (b)Controlado CGPR (c) Contro- lador CGPR - Engavetamento	60
48	<i>Ref</i> $\alpha_3 = 0^\circ$ (a) Controlado CGPR	61
49	(a) <i>Ref</i> $\alpha_3 = 40^\circ$ seguido de <i>Ref</i> $\alpha_3 = -40^\circ$ (b) <i>Ref</i> $\alpha_3 = 50^\circ$ seguido de <i>Ref</i> $\alpha_3 = -50^\circ$	61

Sumário

Resumo	xiii
Abstract	xiv
1 Introdução	1
1.1 Motivação	2
1.2 Definição do Problema	4
1.3 Objetivos	6
1.4 Estrutura da Dissertação	6
2 Plataforma Experimental	8
2.1 Introdução	8
2.2 O Automodelo	9
2.3 <i>Hardware</i> Embarcado	10
2.3.1 Sistema de Odometria	13
2.3.2 Controle de Velocidade	16
2.3.3 Potenciômetros	19
2.3.4 Sensor Infra-Vermelho	21
2.3.5 Sistema de Direção	23
2.3.6 <i>Firmware</i>	26
2.4 <i>Hardware</i> de Comunicação	29
2.5 Conclusão	33

3	Controladores <i>Fuzzy</i>	35
3.1	Introdução	35
3.2	Controlador CpPI	37
3.3	Controlador Global	38
3.4	Controlador CGPR	40
3.5	Esforço computacional	48
3.6	Conclusão	49
4	Resultados Experimentais	51
4.1	Introdução	51
4.2	Ajustes em Malha Fechada do CGPR	52
4.2.1	Controlador da Primeira Articulação: <i>Truck-Trailer</i>	53
4.2.2	Controlador da Segunda Articulação: <i>Truck-Trailer-Trailer</i>	54
4.3	Experimentos: Controlador Global Vs CGPR	56
4.3.1	Experimentos I	57
4.3.2	Experimentos II	57
4.3.3	Experimentos III	58
4.3.4	Experimentos IV	60
4.4	Conclusão	61
5	Conclusões	63
5.1	Trabalhos Futuros	64
	Referências	66
	Anexo A – Estacionamento de veículo não articulados	70
	Anexo B – Firmware da Placa de Controle	93
	Anexo C – Algoritmo de Controle	105

Resumo

Este trabalho apresenta desenvolvimentos numa plataforma experimental para ensaios de manobras com veículos ou robôs móveis multi-articulados, também foi desenvolvido um algoritmo *fuzzy* de controle de configuração e feita a sua análise de desempenho relativamente a outros algoritmos da literatura. A plataforma desenvolvida é compreendida por um veículo multi-articulado em escala, composto por um elemento trator e dois elementos passivos ou *trailers*. Foram confeccionados e embarcados alguns circuitos eletrônicos responsáveis pelo acionamento, sensoriamento e comunicação sem fio do veículo com o computador remoto, onde os algoritmos de controle são processados. Foram descritas as principais soluções encontradas para os problemas enfrentados durante o desenvolvimento da plataforma, sobretudo os de ordem mecânica e eletrônica. O problema de navegação e manobras para trás em veículos multi-articulados é bastante complexo e, portanto, de difícil solução. Uma parte desse problema é o controle de configuração, que possui como objetivo o alcance e manutenção de determinados ângulos pelas articulações do veículo sem que ocorra a situação de engavetamento ou *jackknife*. Assim, como contribuição de controle, foi proposto um controlador de configuração, nomeado *CGPR*, que se baseia na teoria *fuzzy* e utiliza, de forma original, o conceito de ângulo de giro para melhorar o seu desempenho. Foram abordados aspectos relativos à dinâmica e limitações dos movimentos para trás como ângulo crítico, atrasos de resposta e configuração crítica. Por fim, uma versão do *CGPR* com compensadores de malha fechada PID foi implementada e testada na plataforma experimental desenvolvida, onde teve seus resultados comparados aos obtidos por outro controlador.

Abstract

In this work, an experimental platform was developed for testing maneuvers by vehicles or mobile robots multi-jointed, has also developed a fuzzy control algorithm and configuration made its analysis of performance against other algorithms in the literature. The platform built is comprised of an articulated vehicle multi-scale, composed of one element and two tractor trailers or passive elements. Were made and shipped some electronic circuitry responsible for driving, sensing and wireless communication vehicle with the remote computer where the control algorithms are processed. We described the main solutions to the problems faced during the development of the platform, especially those of a mechanical and electronics. The problem of navigation and maneuvering back into multi-articulated vehicles is very complex and therefore difficult to solve. A part of this problem is the control setup, which has as objective the achievement and maintenance of certain angles the joints of the vehicle without causing the situation of jackknife. Thus, as a contribution of control, we propose a controller configuration, named CGPR, which is based on fuzzy theory and uses the original form, the concept of angle of rotation to improve their performance. Were approach aspects of the dynamics and limitations of movements back as critical angle, and response delays critical configuration. Finally, a version of CGPR with compensating closed loop PID has been implemented and tested in experimental platform developed, which had their results compared to those obtained by other controllers.

1 *Introdução*

Os resultados das pesquisas na área da robótica estão cada vez mais presentes em aplicações cotidianas. A robótica tem como principais objetivos possibilitar que tarefas perigosas, cansativas e de alta complexidade possam ser realizadas por robôs, de maneira mais precisa, segura, rápida e com menor custo.

No universo dos robôs com aplicações práticas alguns merecem destaque, sendo eles: robôs manipuladores, amplamente empregados na indústria, robôs submarinos ou ROV (Remotely Operated Vehicle), com uso bastante difundido em tarefas relacionadas à exploração marítima de petróleo, robôs aéreos ou UAV (Unmanned Aerial Vehicle) comum em aplicações militares e os robôs móveis terrestres que apresentam uma vasta gama de aplicações.

Nesse trabalho, serão abordados aspectos relativos à os robôs móveis terrestres. Na sua maioria, esses robôs são capazes de realizar tarefas através da navegação em ambientes muitas vezes desconhecidos à priori. Em geral, a autonomia dos robôs móveis é conseguida através da fusão de informações de sensores proprioceptivos e de detecção do entorno que são processadas por algoritmos "inteligentes", que determinam a ação a ser executada pelo robô.

Dentre os diferentes tipos de robôs móveis, os multi-articulados são objeto de pesquisa nessa dissertação. Esse tipo de robô possui semelhanças, sobretudo em sua cadeia cinemática, com os veículos multi-articulados, que são amplamente empregados no transporte de cargas [2] em diversos locais, como pátios de armazenamento, aeroportos, rodovias e portos, como mostra a Figura 1. De modo similar aos robôs móveis articulados, esses veículos possuem características de movimentação peculiares que restringem a sua utilização. Para reduzir essas limitações as ferramentas de automação disponíveis na robótica móvel podem ser utilizadas para auxiliar o homem na realização de tarefas e manobras complexas com veículos multi-articulados e, desta forma, ampliando ainda mais o seu uso.



Figura 1: Exemplo do uso de veículo multi-Articulado no transporte de cargas.

1.1 Motivação

O crescimento industrial ocorrido nas ultimas décadas, aliado ao avanço dos meios de comunicação, possibilitaram o surgimento de novos mercados consumidores dos mais variados tipos de produtos, que são originados de diversas partes do mundo. Como consequência desse crescimento e globalização do mercado surge uma crescente demanda por soluções inovadoras no ramo da logística de transportes de cargas. Sendo portanto, uma área estratégica e alvo de diversos estudos que visam otimizar custos de estocagem e transporte, minimizar o espaço físico dos pátios de armazenamento e reduzir o tempo de entrega das mercadorias.

Nesse contexto o transporte terrestre, sobretudo o rodoviário, responde por grande parte do volume total de cagas transportas em todo mundo [3]. No Brasil, o transporte rodoviário também é fortemente utilizado, sendo que 60% do PIB (Produto Interno Bruto) brasileiro é transportado por rodovias [4].

Objetivando-se a redução do custo por tonelada transportada, é comum o carregamento de caminhões além da sua capacidade nominal de carga. Essa prática ilegal, a médio prazo aumenta os custos com manutenção do veículo além de provocar o desgaste prematuro das rodovias, prejudicando o transporte terrestre em geral. Como solução para esse problema, o uso de veículos multi-articulados, ou seja, caminhões com um ou mais semi-reboques, é cada vez mais frequente no transporte de cargas. A utilização desse tipo

de veículo, reduz o peso por eixo transportado, diminuindo a degradação das estradas e os custos com manutenção e, dessa forma, aumentando a rentabilidade por tonelada transportada [5].

A Figura 2 mostra veículos com variados números de semi-reboques, conhecidos como *trailers* ou elementos passivos. Com aumento do número de *trailers*, esses veículos passam a ser conhecidos como Rodo Trens, Figuras 2(c) e 2(d), e são empregados em diversos segmentos do transporte terrestre.



(a)



(b)



(c)



(d)

Figura 2: (a) Caminhão não articulado (b) Veículo biarticulado (c) Rodo Trem com quatro *trailers* (d) Rodo Trem com seis *trailers*

A utilização dos veículos multi-articulados traz junto com a diminuição do custo de transporte, uma desvantagem, que é a complexidade de condução. A complexidade das manobras se dá em movimentos para trás, e neste caso, o problema se compara ao de um sistema de pêndulos invertidos múltiplos na horizontal. Muitas vezes, manobras à ré nesses veículos não são executadas nem mesmo por motoristas experientes. Em geral é realizado o desengate dos elementos passivos para a realização de manobras, o que não é prático.

Nos veículos multi-articulados, as manobras são realizadas exclusivamente por meio

das rodas de direção presentes no elemento trator ou *truck*. A cada novo elemento passivo conectado à composição diminui a capacidade humana de produzir ações de controle da direção de modo correto. Isso acontece porque os efeitos da intervenção realizada na direção são propagados e modificados ao longo da cadeia cinemática, não permitindo que o último elemento seja controlado com eficiência.

Durante a movimentação para trás, os veículos multi-articulados podem assumir uma configuração de engavetamento, ou *jackknife*, por lembrar o movimento realizado por um canivete ao ser fechado. Essa condição impede a continuidade do movimento e pode causar danos a estrutura do veículo, necessitando de soluções que a evite.

Diante do exposto, para se aumentar os benefícios do uso de veículos multi-articulados, é de grande importância a existência de ferramentas que auxiliem o motorista ou até mesmo que executem automaticamente as manobras em movimentos à ré nesses veículos. Essa ferramenta traz benefícios como, aumento de segurança, possibilidade de realização de manobras em situações com restrição de espaço e ampliação das aplicações dos veículo multi-articulados como em armazéns automáticos.

1.2 Definição do Problema

Veículos e robôs móveis multi-articulados (V/RMA) são compostos por um elemento trator, responsável pelo sistema de direção e tração, conectado, através de articulações, aos elementos passivos, responsáveis pelo transporte da carga.

O problema de navegação dos V/RMA em manobras a ré, foi abordado em [3], onde foi dividido em diferentes etapas hierarquicamente dependentes. Nesse trabalho foi admitida uma divisão mais detalhada, sendo:

- Detecção de obstáculos ou vagas para estacionamento. Essa etapa consiste em mapear o entorno do veículo através de sensores ultra sônicos ou *laser* identificando possíveis vagas e obstáculos na área de manobra. O mapeamento do entorno também pode ser realizado por câmeras, como proposto em [13]
- Determinação da trajetória. Após os obstáculos ou vagas terem sido identificados, o próximo passo é o planejamento de uma trajetória a ser seguida para que o desvio ou estacionamento seja realizado com sucesso. Esta etapa poderia ser implementada através de um simulador da cinemática de robôs multi-articulados em conjunto com técnicas de controle já desenvolvidas para veículos móveis não articulados.

- Seguimento da trajetória. Esta etapa se preocupa em controlar a posição e a orientação do último elemento da cadeia articulada do veículo, permitindo que a trajetória calculada seja seguida.
- Controle de configuração. É a etapa que visa controlar o ângulo da última articulação da composição tendo como variável de controle o ângulo da direção do elemento trator, cujo efeito é propagado pelos demais ângulos da cadeia possibilitando a execução da etapa de seguimento da trajetória, pela composição dos movimentos articulares.

Para veículos simples, sem articulações, o problema de navegação é mais simples, tendo diferentes soluções na literatura [6–11]. Para esse caso mais simples, o Anexo A mostra partes do trabalho desenvolvido em [12], que contém contribuições decorrentes das pesquisas dessa dissertação sobre o problema de detecção de vaga, determinação de trajetória e seguimento de trajetória.

Para veículos com múltiplas articulações, a navegação é um problema mais complexo e pouco abordado [3]. O controle de configuração é uma etapa hierarquicamente inferior e, portanto, sua solução tem prioridade em relação às demais etapas. Em busca de soluções para essa etapa, alguns trabalhos foram realizados pelo grupo de pesquisa de robôs móveis multi-articulados do LAI-UFES (Laboratório de Automação Inteligente da Universidade Federal do Espírito Santo), ao qual esse trabalho pertence. Os trabalhos desenvolvidos visaram principalmente o controle de configuração angular dos V/RMA em movimentos para trás, os mais relevantes resultados obtidos estão em [1, 3, 13] e serão explorados no Capítulo 3.

A dificuldade em controlar a configuração dos veículos multi-articulados ocorre porque a variação dos ângulos das articulações desses veículos, quando em movimento para trás, possui um comportamento não linear e de difícil descrição matemática [3, 5]. Assim, técnicas de controle que dispensam o uso de modelos analíticos são bastante utilizadas para o problema de controle de configuração desses veículos, sendo principalmente baseados em aproximadores Neurais [3, 14–16] e *Fuzzy* [3, 5, 13, 17–20].

Portanto, essa dissertação trata do problema de controle de configuração angular de V/RMA em movimentos a ré, em que o objetivo de controle é fazer com que seja alcançada uma dada referência angular para a última articulação do veículo.

1.3 Objetivos

A maior parte dos trabalhos apresentados na literatura foi desenvolvida por meio de simulações computacionais. Apesar da facilidade que esse tipo de abordagem oferece, elas se baseiam em equações e modelos matemáticos que tentam descrever o comportamento do sistema. Para serem viáveis, muitas vezes as modelagens são limitadas e, portanto, não representam a realidade do problema. Além disso, fatores inerentes aos experimentos reais como a presença de atrito, imperfeições mecânicas e distúrbios diversos, na maioria das vezes não estão presentes nos modelos e consequentemente nas simulações. Essa simplificação faz com que as soluções baseadas em modelos teóricos muitas vezes não resolvam o problema na aplicação prática.

Das pesquisas que utilizaram protótipos experimentais [21–24], a maioria foi realizada com veículos especialmente construídos e, dessa forma, possuindo comportamento cinemático diferente do encontrado em veículos multi-articulados reais, como em caminhões biarticulados. Assim, um dos objetivos desse trabalho é desenvolver uma plataforma experimental em escala de um veículo real, para análise de desempenho e síntese de estratégias e algoritmos de controle de configuração que permitam a navegação autônoma de veículos móveis multi-articulados.

Apesar dos avanços obtidos nos trabalhos anteriores com os controladores de configuração CpPI [3] e Global [1], a etapa de controle da configuração de V/RMA ainda não foi totalmente desenvolvida. Deste modo, outro objetivo desta tese, é desenvolver um controlador de configuração para V/RMA que apresente vantagens em relação aos anteriormente propostos, tendo a sua validade confirmada por meio de experimentos realizados na plataforma desenvolvida.

1.4 Estrutura da Dissertação

De modo a expor toda a proposta e os trabalhos realizados, essa dissertação foi dividida em 5 capítulos. A seguir cada capítulo é brevemente descrito.

- Capítulo 1: Introdução

É o capítulo inicial, cujo objetivo é contextualizar o problema, faz referência às pesquisas consideradas mais relevantes sobre o controle de manobras à ré em robôs móveis ou veículos multi-articulados. Além disso, este capítulo também apresenta a motivação e o objetivo do trabalho realizado.

- Capítulo 2: Plataforma Experimental Esse capítulo explica a importância do desenvolvimento da plataforma experimental, descreve as ferramentas utilizadas, os problemas e as soluções encontradas no seu desenvolvimento.

- Capítulo 3: Controladores *Fuzzy*

Discute aspectos relacionados a técnicas de controle inteligente e aborda as soluções existentes para o controle de configuração de V/RMA, denominadas CpPI e Global. Propõe uma nova metodologia de controle nomeada CGPR (Controle no Giro por Propagação de Referência). Além disso descreve aspectos relacionados à dinâmica das articulações em movimento a ré de V/RMA.

- Capítulo 4: Resultados Experimentais

Neste capítulo são apresentados os equipamentos utilizados para a realização dos experimentos. Os primeiros experimentos servem para ajustar o controlador CGPR e os demais mostram os resultados obtidos pelos controladores Global e CGPR, servindo para comparar o desempenho de ambos.

- Capítulo 5: Conclusões

Encerra a dissertação com uma discussão geral do trabalho realizado, cita as principais contribuições, problemas encontrados e faz considerações finais propondo sugestões de trabalhos futuros.

2 *Plataforma Experimental*

2.1 Introdução

No estudo dos movimentos, sobretudo à ré, de veículos ou robôs móveis multi-articulados, a existência de uma plataforma experimental é de grande importância, pois facilita o entendimento da dinâmica do sistema proporcionando a geração do conhecimento especialista necessário para a implementação de algumas técnicas de controle, como as baseadas na teoria *fuzzy*. Além disso, uma plataforma permite o desenvolvimento, a validação e a análise de desempenho de soluções de controle, que podem ser obtidas por diferentes métodos e estratégias.

Uma característica útil de experimentos com protótipos reais é que eles incorporam ao teste interferências e distúrbios externos como por exemplo, ruídos, atrito, deslizamento e imperfeições mecânicas, que nem sempre podem ser considerados em modelos teóricos. Isso faz dos experimentos em protótipos reais uma ferramenta muito importante na avaliação de aspectos relacionados a robustez e estabilidade das soluções propostas antes da sua implementação em escala real ou aplicação final.

Em trabalhos anteriores [15], algumas tentativas de construção de um suporte experimental foram realizadas. Contudo, tais tentativas não obtiveram êxito, principalmente devido às dificuldades relacionadas ao projeto mecânico e pela crítica tarefa de fabricação de peças. Uma solução encontrada para esses problemas foi a de utilizar um automodelo industrializado como base para a plataforma experimental.

Desta forma, a plataforma foi desenvolvida em um automodelo do tipo *truck-trailer-trailer*, que possui uma configuração típica de veículos multi-articulados utilizados no transporte de cargas [25]. Ao mesmo tempo, essa configuração contendo três elementos, é suficiente para manter a generalidade do problema de manobras. Além disso, um maior número de elementos passivos poderia inviabilizar testes em ambientes espacialmente restritos, como acontece em experimentos baseados em visão computacional, onde o alcance

da câmera é o limitador da área de movimentação do veículo, conforme descrito em [13].

Uma vez que o automodelo adquirido era desprovido de funções de acionamento, sensoriamento e comunicação de dados, iniciou-se o planejamento e o desenvolvimento dos mecanismos e circuito eletrônicos necessários para dotar o veículo de tais funções. Na Seção 2.2 serão detalhadas as especificações do automodelo utilizado. Na Seção 2.3 são descritas as principais características da eletrônica embarcada, como atuadores e sensores proprioceptivos e finalmente na Seção 2.4 é apresentado o sistema de comunicação sem fio desenvolvido, responsável pela comunicação de dados entre o computador remoto e a plataforma experimental.

2.2 O Automodelo

Conforme dito anteriormente, o automodelo multi-articulado é constituído pelo elemento trato e por dois semi-reboques, medindo no total $1,73\text{ m}$ quando alinhado. Fabricado pela empresa Tamiya® [26] Co. Ltd., o automodelo é uma réplica na escala de 1/14 do caminhão *Globe Liner* e possui como característica uma estrutura mecânica resistente.



Figura 3: Automodelo *Globe Liner*, elemento trator.

O elemento trator é dotado de um sistema de transferência de tração do tipo diferencial e de suspensão independente com feixe de molas e amortecedores, o que facilita o transporte extra de peso sem prejudicar sua movimentação, sobretudo em pisos irregulares. A Figura 3 mostra o elemento trator do automodelo utilizado.

A tração do veículo é do tipo 6X4, ou seja, quatro das seis rodas são tracionadas e é suprida por um motor elétrico *C.C.* da série 540. Acoplado ao motor, existe uma caixa de

redução, ou câmbio, que possibilita a escolha da relação de engrenagens entre os valores de 1 : 32.4, 1 : 17.7 e 1 : 10.6, representando a primeira, segunda e terceira marchas, respectivamente.

Para maior praticidade no manuseio e melhor acomodação do hardware embarcado, a carenagem do elemento trator foi removida. A Figura 4 mostra o veículo completamente montado.



Figura 4: Veículo multi-articulado montado

2.3 Hardware Embarcado

Para o acionamento do mecanismo de direção, controle de velocidade, seleção de marcha do câmbio, conversão de sinais analógicos em digitais, comunicação, temporização de eventos e armazenamento de dados, foi desenvolvido um circuito eletrônico microcontrolado.

Neste ponto é importante ressaltar que o *hardware* embarcado desenvolvido não possui a finalidade de comportar algoritmos de controle de manobras do veículo. Desta forma, as possíveis soluções de controle não estariam restringidas à capacidade de memória, precisão e velocidade de processamento desse *hardware* embarcado.

O controle de alto nível é realizado de modo *off-board*, ou seja, todo o processamento mais complexo destinado ao controle de manobras e determinação de trajetórias pode ser convenientemente executado por um outro computador de maior desempenho e melhores recursos computacionais, que ao final, transmite a ação de controle ao circuito de acionamento no veículo.

O microcontrolador utilizado no projeto foi o *Atmega32* do fabricante *Atmel* [27]. Suas principais características são mostradas na Tabela 1. O uso desse microcontrolador é favorável ao desenvolvimento de sistemas embarcados, pois possui baixo consumo energético e maior desempenho, quando comparado a outros microcontroladores da mesma categoria.

De modo geral, os critérios de escolha dos componentes eletrônicos foram baseados no desempenho, baixo consumo, baixo custo e facilidade de obtenção, sendo conseqüentemente, de fácil reposição.

Tabela 1: Principais características do Atmega32.

Característica	Quantidade
Frequencia de processamento	16 <i>Mhz</i>
Memória de programa	32 <i>Kbyte</i>
Memória RAM	2 <i>Kbytes</i>
Memória EEPROM	1 <i>Kbyte</i>
Saida PWM	4
Entrada Analógica	8
USART	1
I2C	1
Alimentação	3.0 – 5.5 <i>volts</i>

Objetivando-se a expansão da capacidade de armazenamento de dados para valores além da existente no próprio microcontrolador, instalou-se um C.I.(Circuito Integrado) de memória EEPROM(*Electrically-Erasable Programmable Read-Only Memory*) com capacidade de armazenar até 64 *Kbytes*. Esse C.I. é caracterizado por reter a informação mesmo sob a ausência de alimentação. Além disso, ele se comunica com o microcontrolador através de um barramento de comunicação de dados denominado *I2C*(*2C Inter-Integrated Circuit*) que permite a comunicação com mais de um C.I., facilitando futuras expansões de memória.

A expansão da memória embarcada é importante em experimentos, como os que foram realizados em [1] e [14] onde se coletou os valores dos ângulos entre as articulações em diferentes movimentos para trás.

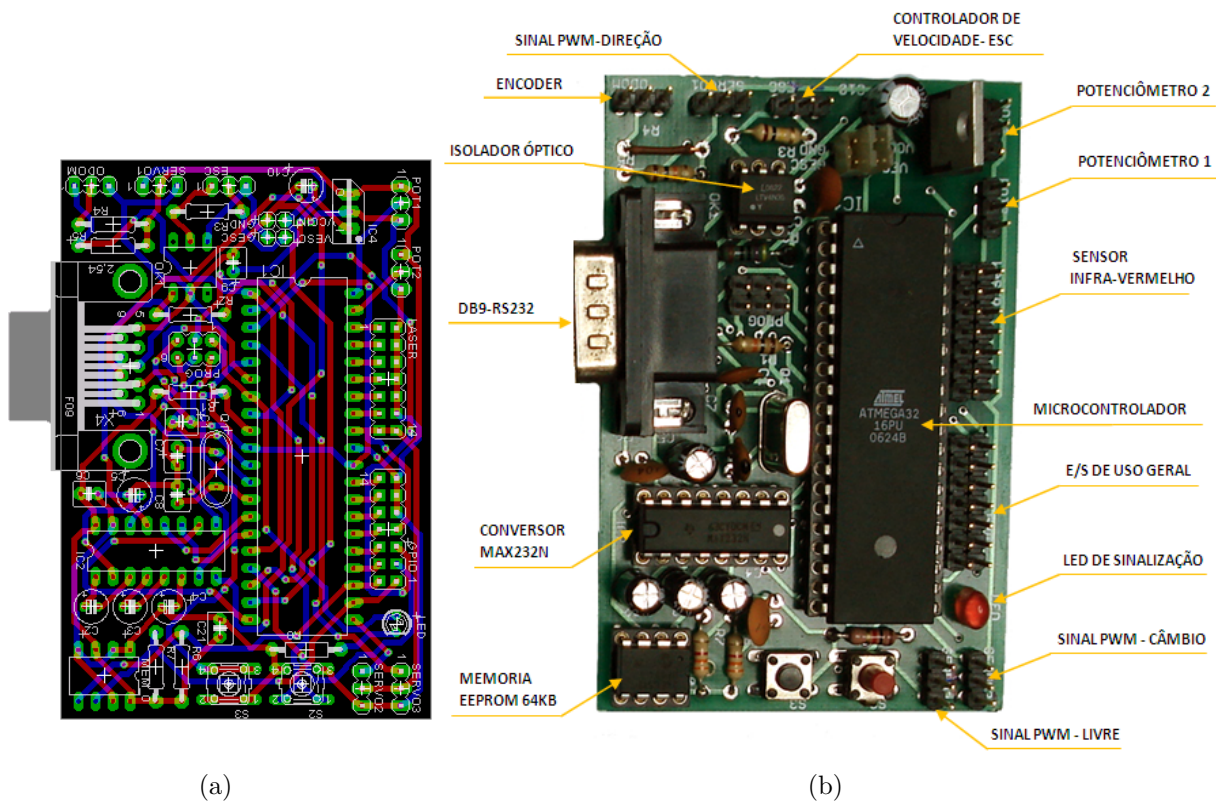
A armazenagem *on-board* de dados, além de ser mais rápida, permitindo a obtenção de uma taxa de amostragem maior, é mais segura, pois evita a necessidade da transferência dos dados coletados para o computador remoto durante o experimento. Assim, ao término dos experimentos, os dados ficam permanentemente armazenados na memória e então podem ser convenientemente transferidos. As especificações da placa de controle são dadas na Tabela 2.

Os projetos das placas de circuito impresso foram desenvolvidos utilizando-se o *software* Eagle Light® [28], programa de C.A.D. (*Computer-Aided Design*) específico para essa finalidade. A Figura 5(a) mostra o desenho em C.A.D. da placa de controle desenvolvida e a figura 5(b) exhibe uma versão montada com a indicação dos principais

Tabela 2: Especificações da placa de controle.

Característica	Quantidade
Dimensões	83X55mm
Entradas/Saídas Digitais	9
Saídas PWM	4
Entradas Analógicas	8
Comunicação	1 U.S.A.R.T. e 1 I2C
Alimentação	15 – 5 VCC
Consumo Médio	23mA

componentes.

**Figura 5:** (a)Layout da placa de circuito (b) Placa devidamente montada.

O circuito eletrônico foi projetado para que pudesse ser alimentado pela bateria que aciona o motor ou por uma outra bateria específica para isso. No entanto, optou-se por alimentar todo o circuito pela bateria utilizada no acionamento do motor. Isso evitou a instalação de uma segunda bateria, reduzindo-se assim, peso e espaço ocupado no veículo. Essa decisão não comprometeu a autonomia no tempo de funcionamento do protótipo, uma vez que todo o circuito eletrônico consome em média 40 mA, valor pouco representativo quando comparado aos 2A de consumo médio do motor. Assim, toda a energia elétrica

do sistema é provida por uma bateria recarregável de NiMh (Níquel-Hidreto Metálico) com tensão nominal de $7.2volts$ e capacidade de $2.500mAh$, conferindo uma autonomia média de 25 minutos de operação ao veículo.

Para se fazer uso dos benefícios desse esquema de alimentação, algumas precauções foram tomadas, como a instalação de filtros capacitivos e isoladores ópticos. Sem essas medidas, os ruídos elétricos gerados pelo funcionamento e acionamento do motor, que é realizado por PWM (*Pulse Width Modulation*), poderiam comprometer o correto funcionamento de todo o circuito eletrônico.

Outro fato importante, que se refere à segurança do sistema, é o de que a bateria, ao se descarregar, inicialmente interrompe o acionamento do motor, permanece com carga suficiente para manter o circuito de controle energizado. Isso evita a situação desfavorável de se possuir tração nas rodas sem o circuito de controle estar alimentado.

A Figura 6, mostra o esquema de ligação entre os diversos periféricos controlados e monitorados pela placa desenvolvida, sendo as principais funções a leitura da odometria, controle de velocidade, leitura dos potenciômetros, leitura de sensores infra-vermelhos, acionamento da direção e comunicação de dados. O funcionamento de cada uma dessas partes será melhor apresentado nas Sub-Seções seguintes.

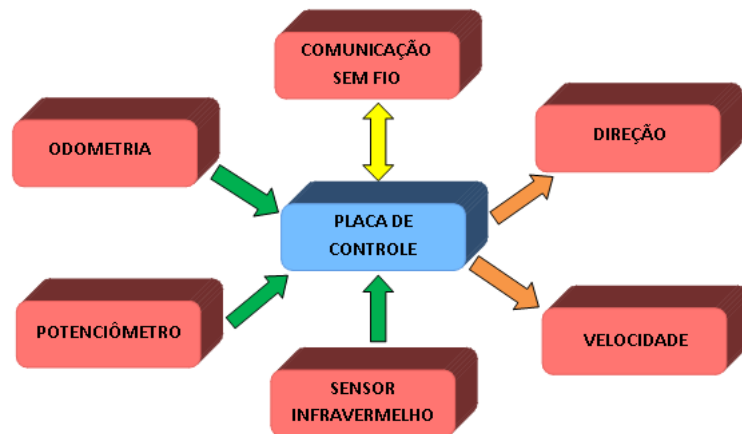


Figura 6: Esquema de ligação entre a placa de controle e os periféricos.

2.3.1 Sistema de Odometria

O sistema de odometria é uma ferramenta muito utilizada na robótica móvel para a estimação da posição do robô [29]. Seu princípio de funcionamento baseia-se na aquisição e processamento de pulsos elétricos gerados por *encoders*. Esses por sua vez, são ele-

mentos eletromecânicos, capazes de detectar movimentos no eixos de rotação onde estão instalados, convertendo-os em pulsos elétricos.

Uma das grandezas que caracterizam o *encoder* é a relação *pulsos/volta*, quanto maior a relação maior é a resolução do *encoder*. Dentre os diversos tipos de *encoder* existentes, destacam-se os com quadratura, Figura 7(c), onde é possível se determinar o sentido de giro através do defasamento de 90° entre dois sinais de pulsos, os absolutos com codificação *Gray*, Figura 7(b), capazes de fornecer a posição do eixo e, por fim, o *encoder* incremental, mostrado na Figura 7(c), que são mais simples e somente detectam o movimento do eixo.

Através do *encoder* é possível se determinar a relação entre pulsos elétricos e voltas no eixo de rotação, e assim, pode-se obter a distância percorrida pelo robô móvel e consequentemente a sua posição. Quando se considera a quantidade de pulsos gerados em um determinado intervalo de tempo, determinar-se a velocidade do móvel.

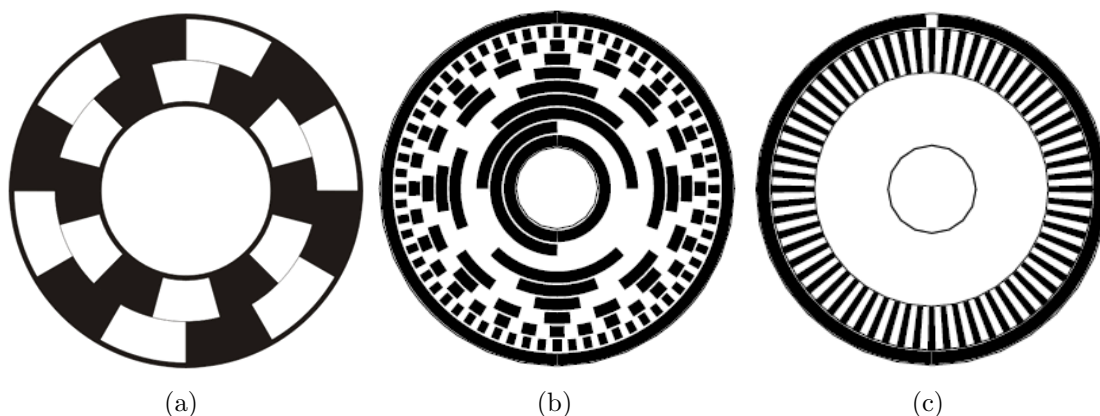


Figura 7: (a) Quadratura (b) Absoluto com codificação *Gray* (c) Incremental.

Neste trabalho foi construído um *encoder* óptico do tipo incremental que possibilitou a determinação da distância percorrida e da velocidade do veículo. O sinal de velocidade é utilizado na realimentação da malha de controle de velocidade, conforme será visto mais adiante.

O *encoder* confeccionado se baseia no princípio da reflexão da luz. Um fototransistor é excitado com a presença de luz, assim, utilizando-se um disco com partes claras e escuras, como mostrado na Figura 7, é possível gerar um sinal de luz pulsado, que altera o modo de operação do fototransistor entre as regiões de corte e saturação, gerando um sinal elétrico pulsado correspondente.

Este *encoder* obteve uma relação de 64 pulsos por volta, maior valor possível alcançado dentre várias tentativas realizadas. Tal limite se deve ao fato de que ao se aumentar o

número de faixas do disco, e automaticamente diminuindo-se a largura de cada uma delas, uma vez que o diâmetro do disco estava definido, notou-se que o fototransistor não conseguia operar nas regiões de corte e saturação e, portanto, não funcionando como chave.

Devido a imperfeições de montagem, o sinal gerado pelo *encoder* apresentava oscilações no valor da amplitude máxima e mínima, o que ocasionou erro na interpretação desse sinal pelo microcontrolador, além disso, as condições de iluminação do ambiente afetavam a operação do fototransistor de maneira sensível.

Para a solução desses problemas, utilizou-se um circuito com um amplificador operacional funcionando como comparador, e assim, ajustou-se o limiar de decisão de forma conveniente. Ao final dos ajustes se obteve um sinal mais adequado na saída do circuito. A Figura 8 mostra o esquema de ligação e as formas de onda geradas antes e após o amplificador operacional.

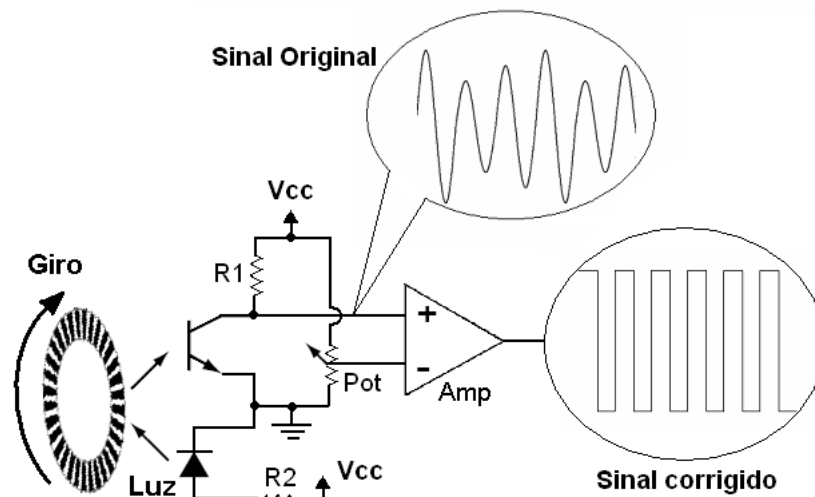


Figura 8: Circuito gerador de pulsos do encoder

Após o sinal ser corrigido pelo amplificador operacional, ele é ligado a uma entrada do microcontrolador, caracterizada por detectar transições de nível lógico. Tal entrada foi configurada para gerar uma interrupção na borda de subida, variação de nível lógico de 0 para 1, e outra na borda de descida, variação de nível lógico de 1 para 0. Dessa maneira, o número de pulsos por volta contados pelo microcontrolador, subiu de 64 (limitado pelo número de raias no disco) para 128, o que melhorou o desempenho do sistema de odometria.

Objetivando-se melhor eficiência, economia de memória e um menor volume de dados

a serem trafegados, a contagem dos pulsos é feita por uma variável de 8 *bits*, assim, contando de 0 a 255. Esse intervalo de contagem é suficiente para que o computador remoto identifique corretamente o estouro do contador, que ocorre quando o valor passa de 255 para 0, e adicione a quantidade 256 a cada estouro para se obter a contagem total dos pulsos. A detecção do estouro se faz observando-se o instante em que o valor da leitura atual dos pulsos é menor do que o valor lido anteriormente.

É importante ressaltar que o local de instalação do *encoder* se deu em um ponto em que o eixo de rotação sofre uma redução de velocidade devido a relação de engrenagens. Para o câmbio posicionado em terceira marcha, a relação entre pulso do *encoder* e a distância percorrida pelo veículo foi de 0,8mm por pulso. A Figura 9 mostra o *encoder* desenvolvido e instalado.

A escolha de funcionamento do sistema em terceira marcha é justificada pelo maior velocidade alcançada no eixo de instalação do *encoder*, gerando portanto mais pulsos para um dado intervalo de tempo, em detrimento do torque, que mesmo reduzido ainda permitiu colocar a composição em movimento.

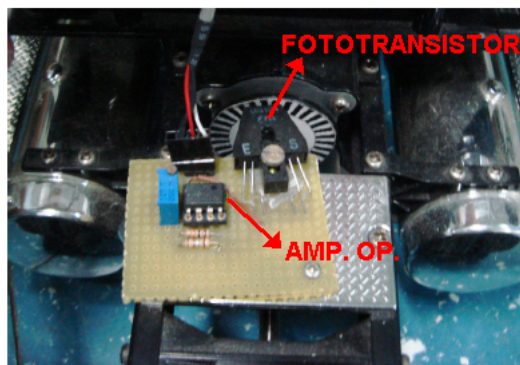


Figura 9: Local de instalação do *encoder*, destaque para o fototransistor e o amplificador operacional.

2.3.2 Controle de Velocidade

Conforme mencionado, a tração do veículo é provida por um motor CC (Corrente Continua). O acionamento do motor é realizado pelo E.S.C. (*Eletronic Speed Controler*), que é um circuito eletrônico *Cmos* responsável por receber o sinal de referência, do tipo PWM, e converte-lo em outro sinal de potência, também PWM, porém com tensão e frequência apropriadas para serem aplicadas ao motor.

Através da variação do ciclo de trabalho do sinal PWM de referência, o E.S.C, por

meio de uma ponte H existente em seu circuito, controla o sentido de giro e a potência fornecida ao motor e, conseqüentemente, variando sua velocidade.

Neste projeto, foi utilizado o E.S.C. *CR230* do fabricante Futaba [30], Figura 10, capaz de operar com tensões de 7.2V a 8.4V e conduzir até 40A em um sentido e 20A no outro sentido da ponte H.



Figura 10: Controlador Eletrônico de Velocidade.

Para o correto funcionamento, o E.S.C. precisa ser calibrado conforme procedimento descrito pelo fabricante. O *firmware* da placa de controle possui a rotina que realiza esse procedimento, como pode ser visto no Anexo B.

A determinação da velocidade é realizada através da contagem do número de pulsos em um intervalo de tempo específico. Esse cálculo é executado por interrupção, o que garante uniformidade nas medidas dos intervalos de tempo, além de permitir que o microcontrolador realize outras tarefas nesse mesmo período.

Com o decorrer dos experimentos e testes realizados na plataforma experimental, observou-se que de acordo com a configuração da cadeia cinemática, as forças existentes entre os elementos da composição ofereciam mais ou menos resistência ao movimento, implicando em uma oscilação ou até mesmo parando a composição.

De acordo com a abordagem de controle adotada, manter a velocidade da composição constante pode ser importante. Pois, para uma dado período de amostragem t a distância percorrida d é dada por $d = v * t$, onde v é a velocidade do veículo. Assim, soluções que inferem o deslocamento d baseadas no tempo, são diretamente influenciados pela velocidade do veículo.

Objetivando-se controlar a velocidade da composição, foi implementado um controlador do tipo PI, sendo executado diretamente pelo microcontrolador embarcado. Os

ganhos proporcional, K_p , e integral, K_i , do controlador foram obtidos analisando-se a resposta do sistema para diferentes valores desses ganhos.

Na Figura 11, que mostra o diagrama do controlador, pode-se observar a presença do elemento saturador, responsável por manter o nível do sinal de controle em uma faixa segura para o sistema.

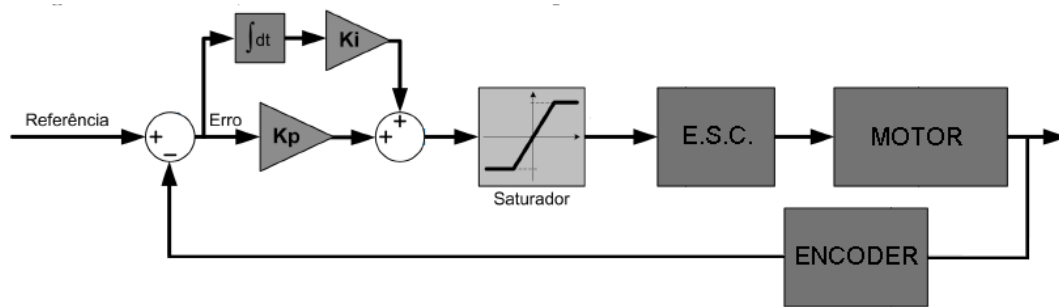


Figura 11: Diagrama de blocos do controlador de velocidade implementado.

A Figura 12 mostra a velocidade da composição com o controlador atuando. Neste exemplo foi dada uma referência de 8 cm/s a partir do repouso. As variações observadas no intervalo entre 11 e 13 segundos mostram a resposta do controlador frente à variação de velocidade sofrida devido a um distúrbio, em seguida, o valor da velocidade retorna para próximo da referência.

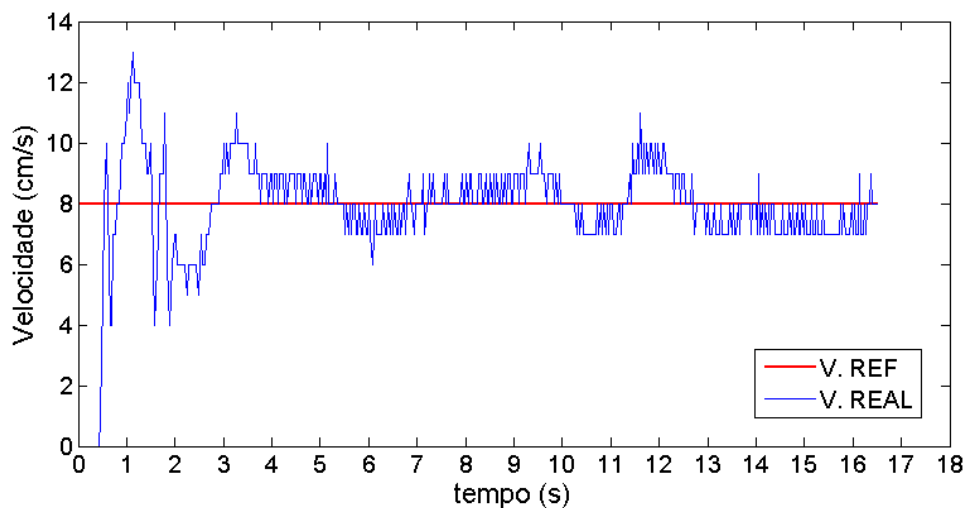


Figura 12: Resposta do controlador de velocidade implementado.

O desempenho do controlador, sobretudo em baixas velocidades, é influenciado diretamente pelo número de pulsos por volta gerados pelo *encoder*. Mesmo com o aumento desse

número, obtido através das medidas citadas, o número de pulsos gerados foi considerado baixo para uma melhor eficiência de controle.

A baixa relação pulsos/volta se deve também ao local no qual foi instalado o *encoder* que, como citado, sofre uma diminuição de velocidade devido ao sistema de redução das engrenagens. Assim, o *encoder* deveria ser instalado no eixo do motor, local onde se obteria uma maior velocidade angular e portanto mais pulsos por volta. Contudo, por restrições mecânicas, não foi possível a instalação neste ponto do protótipo.

2.3.3 Potenciômetros

As variáveis que caracterizam a configuração da composição são os ângulos formados entre os seus elementos. Tais ângulos podem ser obtidos de diferentes formas, como por exemplo, por visão computacional [13] ou através de potenciômetros instalados nas articulações. O uso de potenciômetros é relativamente simples, de rápida leitura e de boa precisão. Para eficácia na estimativa de ângulos através do uso de potenciômetros, esses devem possuir características como linearidade em toda faixa de excursão e precisão adequada.

Atendendo as estes requisitos, foram acoplados dois potenciômetros lineares, um em cada articulação, com valores nominais de $20K\ ohm$ e precisão de 1%. Tais potenciômetros foram cedidos e instalados pela empresa Automática Tecnologia [31], que também desenvolveu um sistema de engate rápido para os *trailers*. A Figura 13 mostra um dos potenciômetros instalados no veículo.

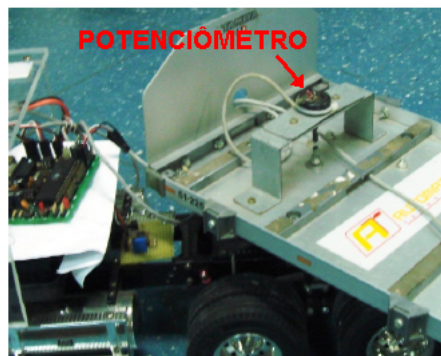


Figura 13: Potenciômetro instalado na junta articulada.

A obtenção dos ângulos das articulações, se dá pela variação da resistência do potenciômetro em função da posição da junta revolvente. A tensão de saída (V_{out}) é dada

por $V_{out} = V_{in} * R_x / R_{pot}$, onde V_{in} é a tensão de alimentação, R_{pot} a resistência nominal do potenciômetro e R_x a resistência associada ao ângulo que se quer medir.

O sinal V_{out} é ligado à entrada de conversão de sinais analógicos para digitais (A/D) do microcontrolador. O conversor possui resolução de 10 *bits*, ou seja, a sensibilidade da medição é de 1/1024 da faixa de variação total do sinal. Desta forma, a relação entre a variação angular por unidade de conversão do A/D é dado por $360^\circ / 1024 = 0,352^\circ$, ou seja, é possível a detecção da variação angular em, no mínimo, $0,352^\circ$.

Um fator importante e que deve ser considerado na conversão digital de grandezas analógicas é a presença de ruído, que altera o sinal a ser convertido e, assim, introduzindo erro na digitalização.

Dentre os principais tipos de ruído, destacam-se os originados por interferências eletromagnéticas, ruído térmico, e especificamente nessa aplicação, os devido a pequenas imperfeições no mecanismo de rotação dos potenciômetros.

No intuito de minimizar os efeitos indesejáveis dos ruídos, os sinais gerados pelos potenciômetros são conduzidos até o microcontrolador por um cabo blindado, além disso, foram instalados filtros capacitivos devidamente especificados. Por fim, o valor final da conversão é obtido após a média dos valores intermediários de cinco conversões seguidas, ou seja, são excluídos o maior e o menor valor, fazendo-se a média do três valores restantes.

Devido ao fato dos potenciômetros não possuírem fim de curso mecânico, vários valores de resistência são permitidos ao se engatar os *trailers*. Dessa forma, é necessária a determinação do valor correspondente ao ângulo zero a cada nova montagem dos elementos da composição nos engates. O valor do ângulo zero é obtido quando se alinha os elementos da composição. A partir desse ponto, os demais ângulos são obtidos através das equações:

$$\alpha_2 = (\text{valor lido1} - \text{valor alinhado1}) * 360 / 1024 \quad (2.1)$$

$$\alpha_3 = (\text{valor lido2} - \text{valor alinhado2}) * 360 / 1024 \quad (2.2)$$

Esse cálculo é realizado no computador remoto, permitindo assim, uma maior precisão do resultado e, ao mesmo tempo, diminuindo o processamento no *hardware* embarcado. Observa-se que os ângulos podem ser positivos ou negativos, o que favorece a identificação da posição relativa entre os elementos da composição.

2.3.4 Sensor Infra-Vermelho

Na robótica móvel é comum o uso de sensores com a finalidade de se obter informações sobre o ambiente que circunda o robô. Tais informações podem ser utilizadas para a detecção de obstáculos ou auxiliar na localização, permitindo que o robô tome uma trajetória adequada. Dentre um grande número de sensores existentes, destacam-se os sensores baseados em raios infravermelho. Tais sensores, são utilizados principalmente para medir a distância entre o robô e obstáculos, ou objetos de referências.

O princípio de medição de distância usado nos sensores infravermelhos desse projeto é o de triangulação. O procedimento de medição, inicia-se com a emissão de um feixe de luz infravermelho pelo sensor, caso exista reflexão da luz por algum corpo, ela retorna e atinge o sensor com um determinado ângulo de incidência.

O ângulo de incidência se relaciona com a distância entre o sensor e o objeto e entre os pontos de emissão e recepção da luz no próprio sensor, que é fixa, conforme ilustrado na Figura 14.

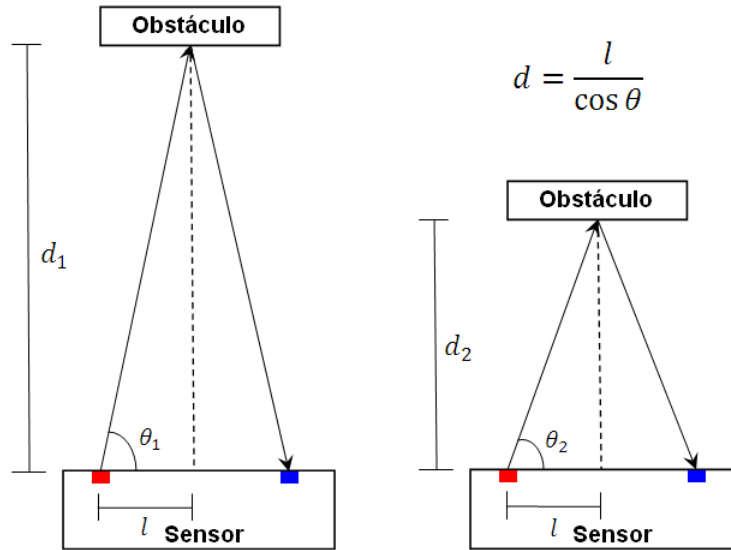


Figura 14: Esquema de Triangulação utilizado pelo sensor infravermelho.

Para permitir a realização de experimentos que necessitam detectar obstáculos e medir distância entre o veículo e algum objeto, conforme realizados em [12], foram adquiridos 4 sensores infravermelhos fabricados pela empresa *SHARP* [32]. Dois sensores são do modelo *GP2D12* e os outros são do modelo *GP2D120*. Esses sensores são capazes de medir distâncias entre 10 – 80 *cm* e 4 – 30 *cm*, respectivamente.

De acordo com suas especificações, esses sensores requerem alimentação de 5 *Volts*

e fornecem como saída um sinal analógico que varia de 0,4 *Volts* a 2,6 *Volts* para o *GP2D12* e de 0,35 *Volts* a 2,8 *Volts* para o *GP2D120*.

Para uma correta medição, esses sensores exigem um distância mínima de posicionamento do objeto. Para distâncias menores que a mínima, a resposta do sensor se torna irregular e imprecisa. A Figura 15 mostra, segundo o fabricante, os gráficos que relacionam a saída em *Volts* com a distância em *cm* para cada sensor.

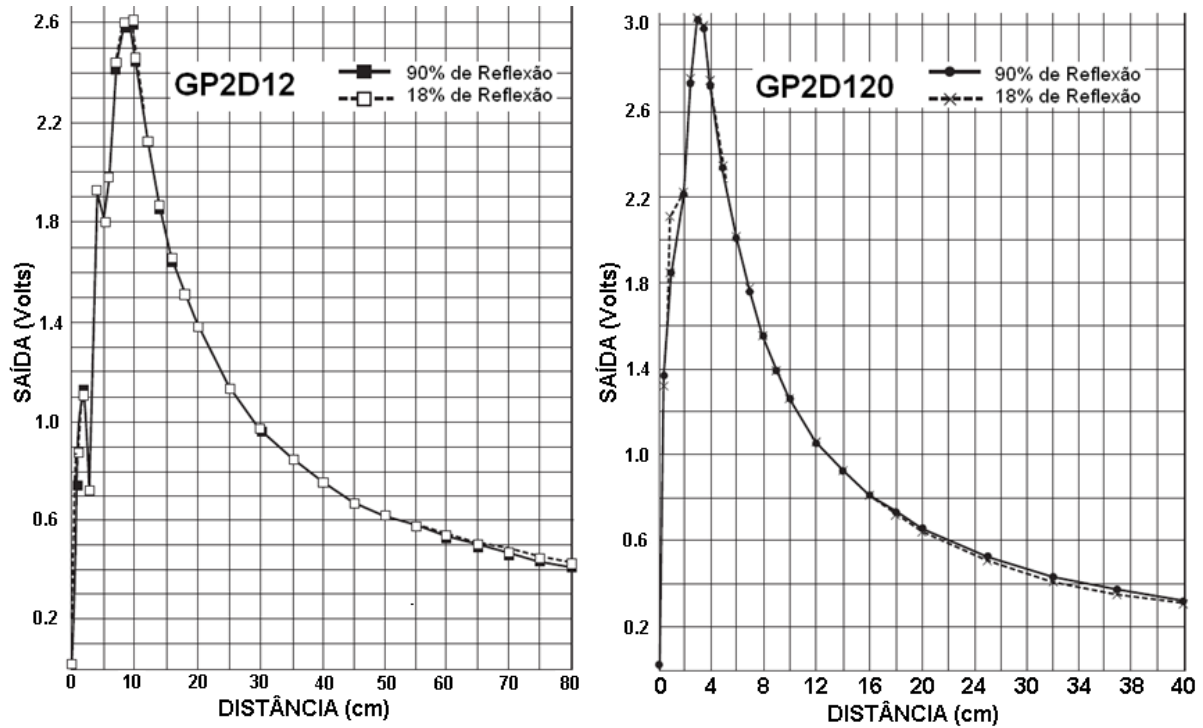


Figura 15: Esquema de Triangulação utilizado pelo sensor infravermelho.

A conversão A/D é realizada pelo *hardware* embarcado, que possui resolução de 10 bits. Após a conversão, os dados são transmitidos para o computador remoto, que então, relaciona o valor recebido com a distância do objeto ao sensor.

É recomendado a calibração do sensor para corrigir eventuais erros de medição. A calibração é feita relacionando-se, para alguns pontos, os valores convertidos de tensão da saída do sensor e a medida real das respectivas distâncias. Valores intermediários podem ser encontrados através de simples interpolação desses pontos. A Figura 16(a) mostra os sensores instalados no elemento trator e o anteparo utilizado para a calibração. A Figura 16(b) mostra a curva de calibração obtida para o sensor *GP2D12*.

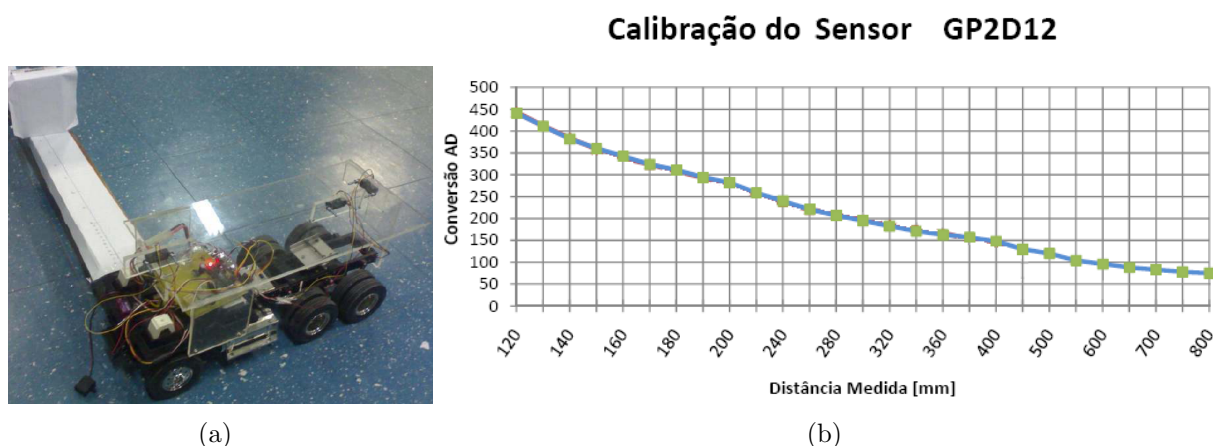


Figura 16: (a) Anteparo de calibração (b) Curva de calibração para o sensor *GP2D12*

2.3.5 Sistema de Direção

O ângulo das rodas de direção é a única variável de controle sobre as manobras dos veículo multi-articulado. Para um desempenho satisfatório do controle do veículo é fundamental que a atuação no mecanismo de direção seja suficientemente rápida e precisa.

Para fornecer tal capacidade ao sistema de direção utilizou-se um servo motor acoplado ao mecanismo de direção, situado na parte inferior do veículo. O servo utilizado foi o *S3003* e suas especificações são encontradas em [30].

O servo motor é um elemento eletro-mecânico capaz de posicionar seu eixo de giro em uma determinada posição, que é definida através de um sinal elétrico de referência. Normalmente, são dispositivos de malha fechada, ou seja, internamente a presença de um circuito de controle PID (controlador proporcional integrativo e derivativo) faz com que a referência seja alcançada com certa rapidez e precisão. A realimentação da posição é feita por um potenciômetro instalado no eixo de giro do servo. Os servos se utilizam de um sistema de engrenagens em que o torque é amplificado, isso permite o uso de motores menores, facilitando a sua miniaturização.

O envio da referência de posição do eixo para o servo é feito por um sinal PWM. Esse sinal se relaciona diretamente com o ângulo requerido para o eixo, conforme mostra a Figura 17.

Sistemas mecânicos são suscetíveis a folgas e deformações presentes nas conexões entre seus diversos componentes. Durante os testes iniciais do sistema de direção do veículo,

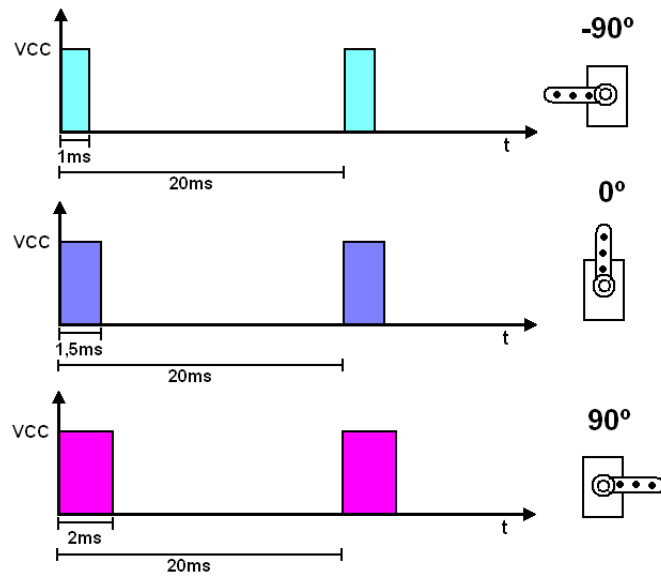


Figura 17: Sinal de controle do servomotor.

percebeu-se a presença de folgas mecânicas. Esse problema acarreta uma não linearidade do sistema de direção e, dessa forma, influenciando na relação entre o sinal de referência e o efetivo ângulo alcançado pelas rodas.

Além disso, outro fator contribuinte para a não linearidade do sistema de direção é o aspecto construtivo do mecanismo de direção, que transmite o movimento de rotação do servo motor às rodas por meio de uma conversão de movimento angular em linear. Esse fato, faz com que de acordo com a posição da direção, ocorram diferentes variações do ângulo da roda para uma mesma variação angular no eixo do servo.

Para uma melhor análise dos problemas apresentados pelo sistema de direção, foi desenvolvido um dispositivo capaz de fornecer a relação entre o sinal de controle aplicado ao servo e o valor real do ângulo alcançado pela direção.

O dispositivo desenvolvido, mostrado na Figura 18, é formado por uma barra em alumínio e possui em suas extremidades um suporte giratório que se encaixa na roda da direção do elemento trator, cada um dos suportes pode girar livremente, sendo que cada um deles está diretamente ligado a um potenciômetro.

Com o veículo posicionado no dispositivo foram enviados comandos ao servo de modo que as rodas atingissem as duas extremidades, direita e esquerda, e na sequência, retornassem à posição inicial. Durante esse percurso os sinais gerados pelos potenciômetros eram digitalizados e salvos para serem posteriormente analisados.

O resultado obtido para o primeiro experimento pode ser observado na Figura19(a), de

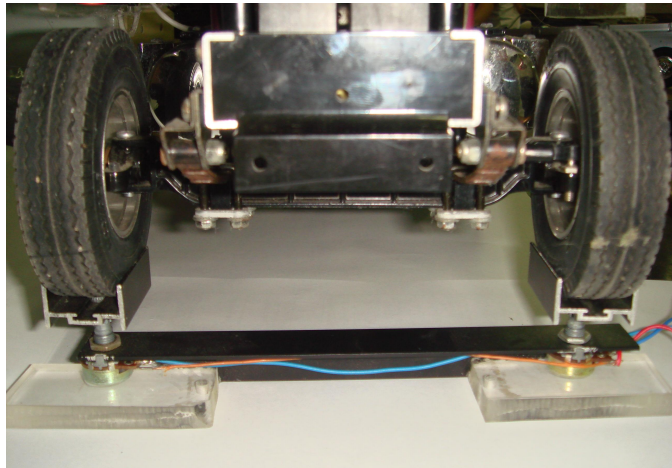


Figura 18: Aparato de calibração da direção.

onde foi possível a quantificação das folgas mecânicas, que possuíam o valor aproximado de 9.7° .

As folgas mecânicas são evidenciadas no momento da mudança do sentido do movimento. Assim, a solução encontrada para o problema foi a de compensar o sinal de controle, através da soma ou subtração de uma constante correspondente ao tamanho da folga. Essa operação é realizada sempre que há uma inversão no sentido de movimento das rodas, ou seja, da direita para esquerda e *vice-versa*.

Essa medida faz com que seja necessário um procedimento de inicialização do sistema de direção a cada vez em que o *hardware* é energizado. Esse procedimento consiste em colocar a direção em uma extremidade e na sequência posiciona-la no centro, desta forma determina-se as condições iniciais para a aplicação da operação descrita.

Com o anteparo desenvolvido foi possível a determinação da constante necessária para a compensação das folgas e não linearidades existentes. As Figuras 19(b) e 19(c) mostram etapas intermediárias desse processo e a Figura 19(d) exhibe a relação final alcançada, na qual se obteve um erro de aproximadamente 1.1° para a posição central.

Cabe ressaltar, que as folgas podem variar de acordo com o piso, pois em locais que oferecem pouco atrito, as deformações e torções são menores quando comparadas às existentes em pisos com maior atrito. Isso acontece principalmente devido à elasticidade das peças plásticas e dos pneus de borracha que compõem o veículo.

Finalmente, é importante ressaltar que esse procedimento permitiu não só a linearização do sistemas de direção mas também a diminuição do erro, que inicialmente era de 9.7° e foi para 1.1° . Isso possibilitou uma melhor operação do sistema de direção,

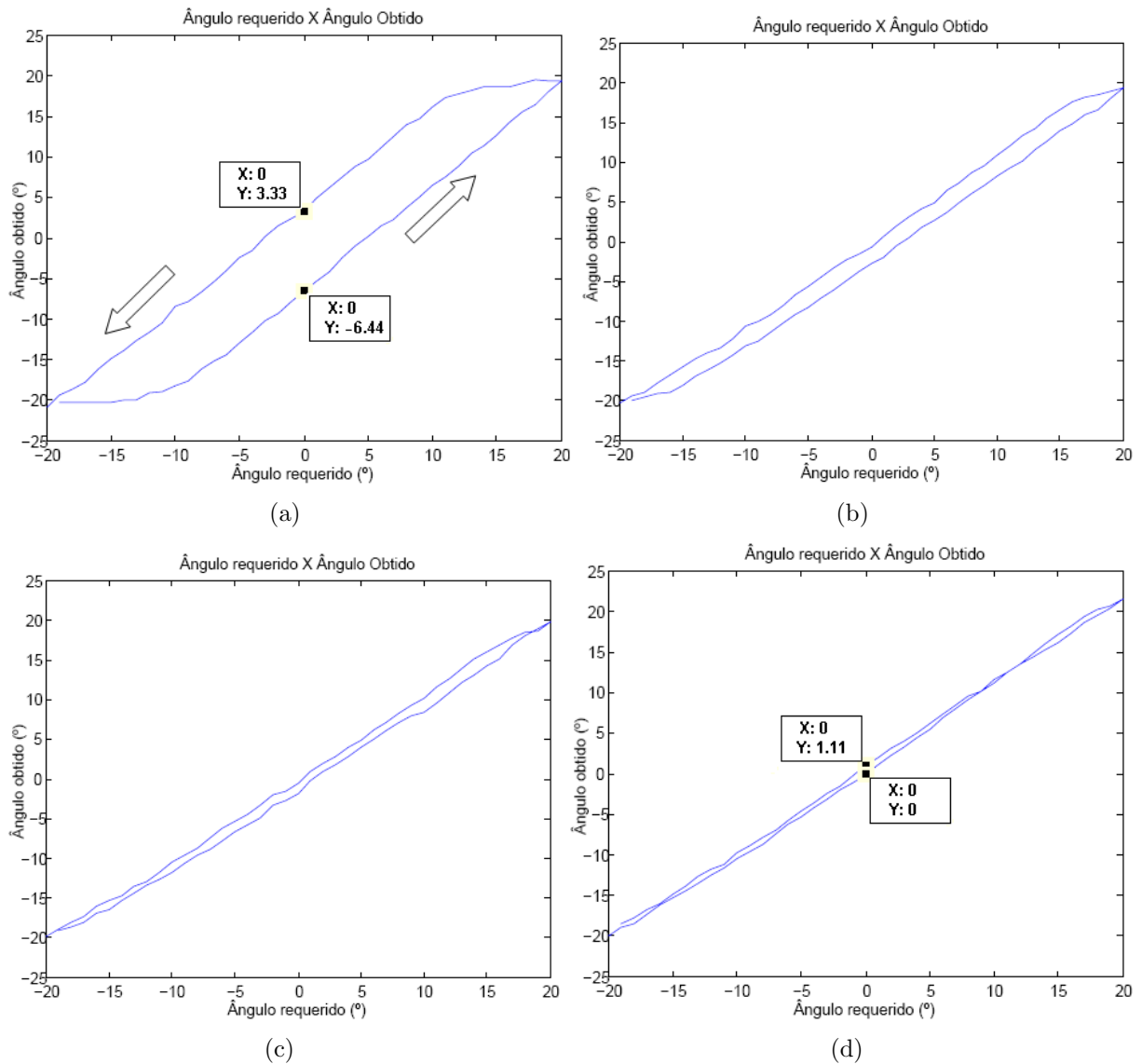


Figura 19: (a) Resposta inicial do sistema de direção (b) Primeira correção
(c) Segunda correção (d) Resposta final do sistema corrigido.

que conforme mencionado, é vital para o controle dos movimentos da composição. Com esse experimento, foi possível a determinação do máximo ângulo alcançado pelas rodas da direção, sendo de $\pm 20^\circ$. Como será visto, esse ângulo está diretamente relacionado à capacidade de manobra da composição.

2.3.6 Firmware

Esta subseção tem por objetivo descrever o protocolo e o software embarcado, ou *Firmware*, desenvolvidos para o projeto, além disso, especifica as ferramentas e os recursos computacionais utilizados.

A escolha da linguagem e da metodologia de programação receberam uma atenção especial nesse trabalho. Tal preocupação se fundamenta na necessidade de que novas funcionalidades possam ser posteriormente implementadas de maneira simples e modular, possibilitando a reutilização dos códigos aqui gerados em trabalhos futuros.

Foi utilizada a linguagem de programação *C* para o desenvolvimento do *firmware*. Essa linguagem possui como grande vantagem em relação as demais, o fato de possuir uma vasta quantidade de compiladores que atendem diversos tipos de microcontroladores, o que facilita a reutilização do código em uma possível migração do *hardware*. Além disso, possui documentação amplamente difundida na literatura e na *internet*. Outra característica importante é o fato de a linguagem *C* ser de alto nível [33], o que favorece a produção rápida de códigos e, ao mesmo tempo, possuir funções comuns a linguagens consideradas de baixo nível, permitindo a geração de códigos otimizados tanto em velocidade de execução quanto em tamanho.

Foi utilizado o compilador *CodeVisionAVR* [34], versão estudante, que possibilita a geração de códigos de até *2KByte* de tamanho. Esse compilador possui suporte para o programador USB MKII [27], que dispensa o uso de alimentação externa, característica desejável, principalmente na realização de experimentos em campo. O Anexo B contém o *firmware* desenvolvido.

Durante a confecção do código, priorizou-se o uso de interrupções para o tratamento e geração de eventos externos e internos à placa de controle. Dessa forma, a detecção e o gerenciamento de tarefas como recepção de dados, contagem de pulsos do *encoder* e geração de sinais *PWM*, dentre outros, foram realizados por meio de recursos existentes no *hardware* do microcontrolador. Esse tipo de abordagem permite a realização de tarefas de forma paralela, o que contribui para a melhoria do desempenho do sistema eletrônico. A Figura 20 mostra o fluxograma do programa construído.

Para que fosse possível controlar, acessar e modificar os valores das diversas variáveis presentes na plataforma experimental, foi desenvolvido um protocolo de comunicação. Esse protocolo é formado basicamente por um caracter (necessariamente minúsculo) identificador da função e, caso exista, o parâmetro da função. A seguir, são mostrados os caracteres *ASCII* que devem ser enviados e a descrição das funções implementadas.

- "n": seleciona velocidade zero para o protótipo, desliga o motor.
- "c": reinicializa todo o sistema embarcado.
- "w": aumenta a velocidade em uma unidade, controle manual de velocidade.

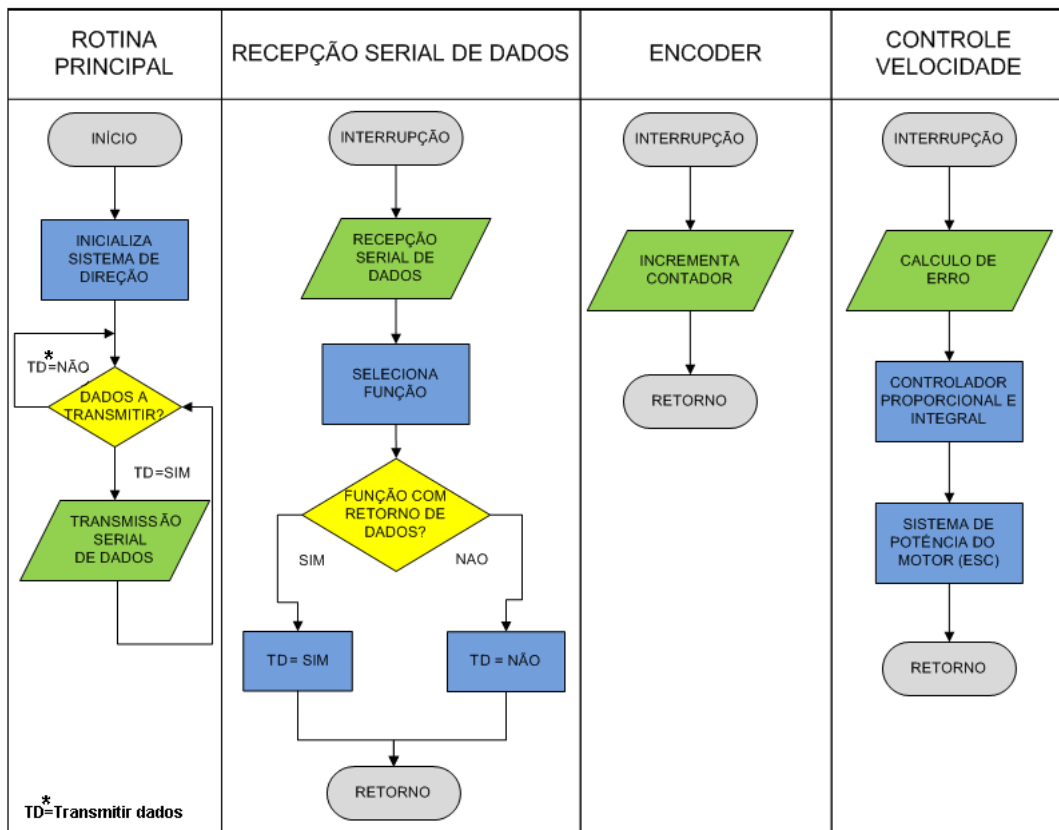


Figura 20: Fluxograma das principais rotinas do *firmware*.

- "s": diminui a velocidade em uma unidade, controle manual de velocidade.
- "o" < *senal* > < *valor* >: Controle automático de velocidade. Caso < *senal* > = + o movimento é para frente e caso < *senal* > = - o movimento é para trás. O parâmetro < *valor* > é um inteiro de 0 a 9 e indica a velocidade.
- "a": varia o ângulo da direção em +1°, sentido anti-horário.
- "d": varia o ângulo da direção em -1°, sentido horário.
- "v" < *senal* > < *valor* >: Define o ângulo da direção das rodas. Caso < *senal* > = + o ângulo é à esquerda e caso < *senal* > = - o ângulo é a direita. O parâmetro < *valor* > é um inteiro de dois dígitos variando de 00 a 20 e indica o valor absoluto do ângulo.
- "t" < *senal* > < *valor* >: Possui a mesma função do "v", porém, retorna o valor dos ângulos das articulações e a contagem do *encoder*.
- "p" < *senal* > < *valor* >: Possui a mesma função do "v", porém, salva no *hardware* o valor dos ângulos das articulações naquele instante.

- "z": Requisita os valores lidos nos potenciômetros.
- "m": Requisita o valor da contagem do *encoder*.
- "k": Requisita os valores lidos nos sensores infra-vermelhos.
- "i" < *valor* >: Inicia a gravação dos ângulos das articulações. O parâmetro < *valor* > é um inteiro de 0 a 9 e indica o intervalo de tempo, em múltiplos de 33ms, entre a gravação dos ângulos.
- "r": Requisita os valores salvos no *hardware* dos ângulos das articulações.

Seguindo esse padrão novas funcionalidades podem ser acrescentadas ao protocolo, sempre que houver necessidade. Para tanto, será necessário a reprogramação do *firmware*.

2.4 Hardware de Comunicação

O desenvolvimento do *hardware* de comunicação, responsável pela transmissão e recepção de dados, exerceu um papel muito importante neste projeto. Tal importância se deve principalmente ao fato de que todo o processamento das informações de configuração e controle das manobras do veículo multi-articulado serem realizados por um programa executado no computador remoto.

O programa de controle de manobras recebe um sinal de referência, que pode ser fornecido por um *joystick* ou inserido no próprio programa, calcula a ação de controle e a transmite para o veículo, esse por sua vez retorna os valores dos ângulos das articulações para o programa de controle de manobras.

Essa arquitetura de processamento remota, exigiu a utilização de um *link* de comunicação de dados confiável e com desempenho satisfatório, sobretudo no que se refere à velocidade de comunicação, que influencia diretamente a atualização do sinal de controle no veículo.

Outra premissa considerada no projeto, foi a de conceber a comunicação de forma modular, ou seja, independente do *hardware* de controle embarcado. Essa medida, visa facilitar uma possível substituição do dispositivo de comunicação, por outros com diferentes tecnologias sem a necessidade de alterações na placa principal de controle.

Inicialmente, no que se refere ao meio físico de propagação, dois modos de comunicação foram levados em consideração, sendo: o que utiliza fios e o que não utiliza fios, conhecidos

como comunicação *wired* e *wireless*, respectivamente.

Apesar de ser normalmente mais confiável que a *wireless*, a comunicação por meio de cabos foi inicialmente descartada, pois esse modo apresenta ao veículo uma maior restrição de manobras e, além disso, o próprio peso e tração dos cabos poderiam alterar a dinâmica do sistema. Desta forma, optou-se pela comunicação sem fio. A Figura 21 ilustra a topologia de comunicação e processamento utilizada.

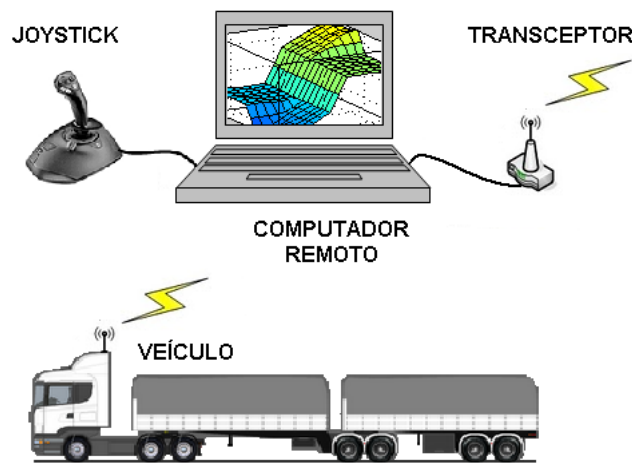


Figura 21: Arquitetura de processamento e comunicação.

Originalmente, a comunicação foi realizada utilizando-se um par de transceptores modelo *FM232c – 433* do fabricante *RF Solutions* [35]. Tais transceptores, objetivavam a realização de um *link* serial do tipo *RS232C* sem fios. Para tal, todos os protocolos e serviços de conexão como, checagem de erro, retransmissão de dados perdidos e identificação do remetente/destinatário já estavam implementados nos próprios rádios.

Na prática, esse rádio mostrou desempenho insatisfatório, principalmente em relação a dois aspectos: taxa de erros e taxa de transmissão. Uma elevada taxa de erros fazia com que a mesma informação fosse retransmitida diversas vezes gerando atrasos de comunicação e prejudicando a atualização do controle. Finalmente, uma baixa taxa de comunicação de dados implicou em uma baixa amostragem de configuração do sistema, assim, comprometendo a velocidade máxima com a qual a composição poderia se mover sob controle.

Um novo modelo de transceptor foi escolhido dentre diversos tipos existentes, tal escolha se deu após análise de características técnicas, disponibilidade no mercado nacional e custo. O modelo escolhido foi o *TRF-2.4Ghz*, Figura 22(a), do fabricante *Laipac Tech* [36]. Suas principais características são a operação na frequência de 2.45Ghz, que pertence

a uma das faixas ISMB (*Industrial Scientific and Medical Band*) definidas no Brasil, além disso, possui taxa de transferência selecionável entre 250kbps e 1Mbps , alimentação em 3V e baixo consumo de energia.

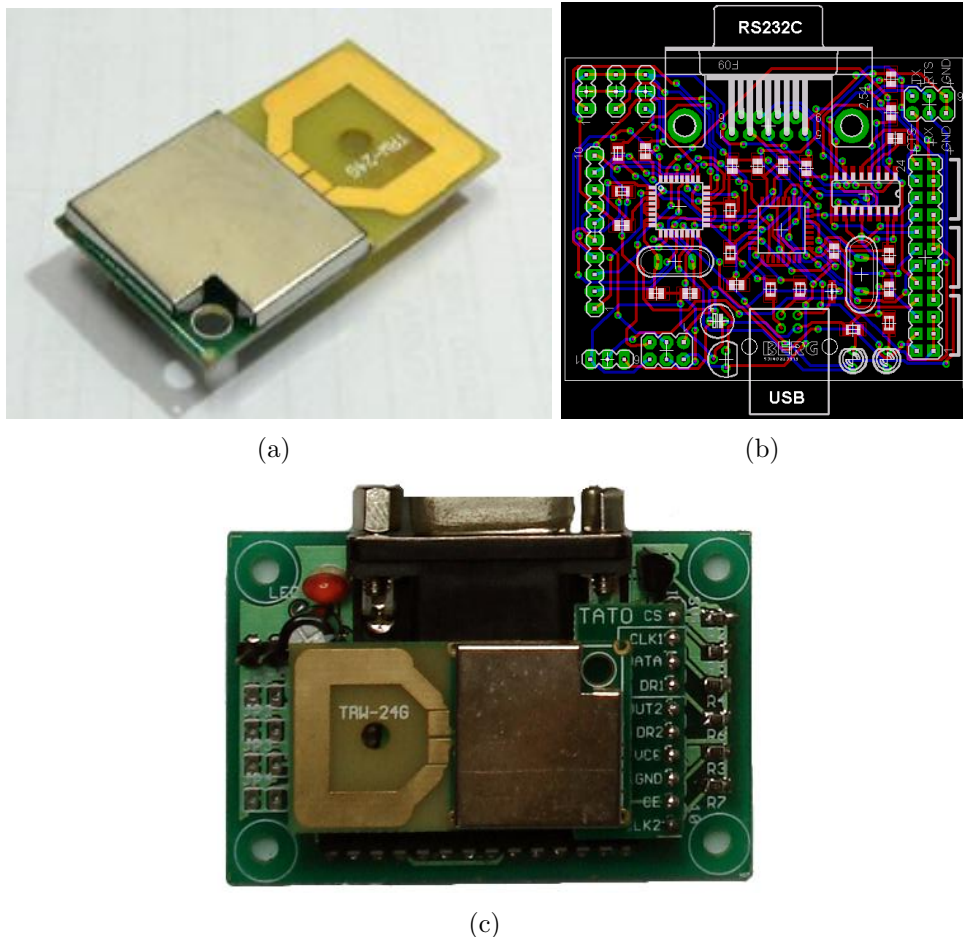


Figura 22: (a) Transceptor TRF 2.4Ghz e (b) Layout da placa de circuito (c) Módulo de comunicação montado.

A figura 22(b) mostra o *layout* da placa de circuito impresso com tecnologia SMD (*Surface Mount Device*) desenvolvida através do *software* Eagle Lite®. Esse circuito é responsável por inicializar, configurar, *bufferizar* dados e controlar o transceptor. Além disso, também possui a função de compatibilizar os níveis lógicos de tensão e protocolos de comunicação entre o elemento final (computador e *hardware* embarcado) e o transceptor. A Figura 22(c) mostra o conjunto, placa de circuito e transceptor, montado.

Após a montagem do módulo de comunicação, foi iniciado o desenvolvimento do seu *firmware*, que foi projetado para prover o sistema de comunicação de serviços relacionadas à garantia de recepção de dados, uma vez que o transceptor é desprovido de tal capacidade. Para tanto, foi implementado um protocolo de comunicação com confirmação de recebimento em 2 vias, além de características como retransmissão de dados em caso

de perda ou falha de comunicação e identificação da aplicação através de senha. Essas medidas ajudaram a evitar interferências nos dados trafegados, atribuindo maior robustez à comunicação. A Figura 23 mostra o datagrama do protocolo desenvolvido, onde cada campo corresponde a um *byte*.

Dado 0	Dado 1	Dado 2	Dado 3	Dado 4	Tam. Dados	N. Seq.	Senha
--------	--------	--------	--------	--------	------------	---------	-------

Figura 23: Datagrama de comunicação.

Onde:

- Dado0-Dado4: Disponíveis para as informação que se deseja trafegar.
- Tam. Dados: indica quantos dos campos de Dado foram utilizados.
- N. Seq.: contém o número de sequência do pacote, permite a identificação do pacote.
- Senha: Contém a senha, caracter que identifica a aplicação.

O protocolo de comunicação desenvolvido introduziu a necessidade de transferência de dados de controle, sendo os três últimos campos do datagrama. Isso acarretou a presença de *Overhead* (dados que não transportam informação propriamente dita) e desse modo, houve uma redução na taxa efetiva de transmissão. Ao final a taxa de transmissão ficou em torno de 20kbps para um distância de 5m entre transmissor e receptor em ambiente fechado. As principais características do módulo de comunicação estão representados na tabela 3.

Tabela 3: Características do módulo de comunicação.

Caractéristica	Quantidade
Dimensões	$48\text{mm} \times 60\text{mm}$
Processador	<i>Atmega8</i>
Buffer Rx/Tx	400Bytes
Interface Serial	<i>USB/RS232C</i>
Taxas de comunicação	$9.6 - 19.2 - 38.4 - 57.6 - 115\text{Kbps}$
Alimentação	5V
Consumo Médio	15mA

Uma característica importante contemplada pelo projeto de comunicação, é o suporte a conexão via interface USB (*Universal Serial Bus*). Esse tipo de interface de comunicação

é muito comum nos computadores atuais, sobretudo em computadores portáteis, onde o uso do padrão serial *RS232* é pouco frequente.

Uma das grandes vantagens do barramento USB é permitir a alimentação do circuito a ele conectado, evitando-se assim, a necessidade de uma fonte de alimentação externa, ou bateria, para o funcionamento desse circuito. Esse recurso foi muito útil em testes realizados com computadores portáteis em ambientes externos. A transferência de dados ocorre como uma comunicação serial com configuração 38400bps , 8 bits , sem paridade e 1 *Stop bit*. A Figura 24 mostra o módulo de comunicação com interface USB.

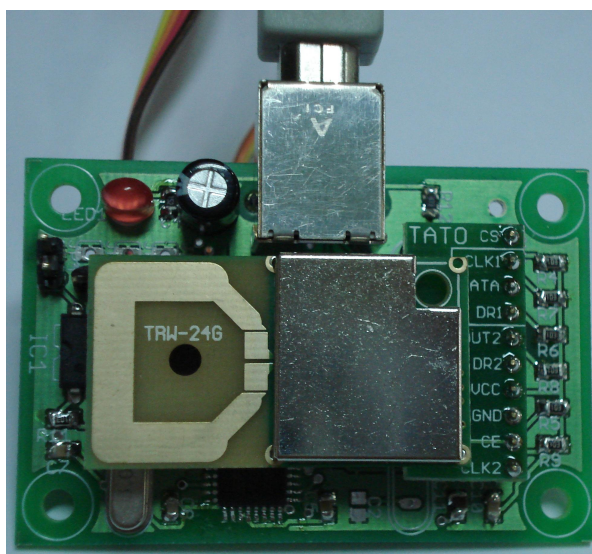


Figura 24: Placa USB conectado ao computador.

Por fim, é importante ressaltar que esta parte do projeto consumiu bastante tempo e esforço, contudo, foi de grande aprendizagem por envolver conhecimentos nas áreas de telecomunicações e eletrônica.

2.5 Conclusão

Em muitas situações reais e de cunho prático o entendimento e o estudo de fenômenos ou problemas relacionados a essas situações apenas são possíveis graças a plataformas de desenvolvimento, onde possíveis soluções e teorias podem ser testadas e comprovadas através de experimentos reais.

A plataforma experimental construída permitiu a realização de diversos trabalhos de pesquisa, teses de mestrado, monografias e artigos científicos. Atualmente, outros estudos estão sendo realizados com o apoio dessa ferramenta.

Nesse sentido, esse capítulo apresentou as principais partes responsáveis pelo acionamento, sensoriamento e controle da plataforma experimental desenvolvida para simulações e validações de estratégias de controle de manobras de veículos móveis multi-articulados. Foram descritos os principais problemas encontrados durante a construção e adaptação do protótipo, explicitando-se as respectivas soluções, sempre tendo em mente o atendimento aos requisitos do problema com o melhor custo/benefício.

Por fim, observa-se que o conhecimento interdisciplinar foi característica presente nessa etapa do trabalho, relacionando conteúdos de automação, robótica, telecomunicações, eletrônica e programação.

3 *Controladores Fuzzy*

3.1 Introdução

Diversas técnicas de controle (clássico, moderno e inteligente) possuem aplicações freqüentes em uma ampla gama de problemas, gerando resultados positivos em diversas áreas da atividade produtiva, sobretudo na industrial, aeroespacial, médica e particularmente no campo da robótica [37].

Em geral, o desenvolvimento dessas técnicas pressupõe as seguintes etapas: modelagem do sistema físico, descrição matemática, análise, projeto e validação do controlador [38]. Porém, em problemas de difícil modelagem ou abordagem analítica, as soluções “*model free*”, denominadas inteligentes, vem sendo cada vez mais utilizadas. Tais soluções são baseadas na aproximação numérica, em geral, via redes ou via regras, da relação entre a entrada e a saída do sistema. As abordagens Neural [39] e *Fuzzy* [40–44] são exemplos de técnicas de controle modernos bastante difundidas.

O problema de manobras à ré de robôs móveis e veículos multi-articulados possui as características de ser não linear e de difícil modelagem. Nesse contexto, as técnicas de controle inteligente podem contribuir para a solução do problema. Nessa direção, diversos trabalhos foram realizados pelo LAI-UFES (Laboratório de Automação Inteligente da Universidade Federal do Espírito Santo) objetivando a solução do problema de manobra, que está organizada em diferentes etapas. As etapas no nível mais elevado são as seguintes: detecção de obstáculos ou vagas, planejamento de trajetória, seguimento de trajetória e o desvio ou estacionamento do veículo, no caso de obstáculo ou vaga respectivamente.

Para a execução das ações de controle hierarquicamente mais elevadas, com desempenho satisfatório é fundamental que o nível abaixo, de controle dos ângulos das articulações do veículo multi-articulado, também apresente desempenho adequado .

O Anexo A se refere ao problema de manobras de um veículo simples num nível mais alto, e esse capítulo ao veículos articulados no nível mais baixo. Este segundo caso,

denominado controle de configuração, consiste em fazer com que uma dada referência, para o ângulo da ultima articulação, seja alcançada e mantida durante o movimento para trás. Para tanto, tem-se como única variável de controle, o posicionamento das rodas de direção.

Com esse objetivo em mente, o desenvolvimento da solução de controle aqui proposta se dá de forma geral, ou seja, para um veículo multi-articulado com n articulações e, portanto, com n graus de liberdade. A implementação e a validação do controle é realizada na plataforma experimental apresentada no capítulo anterior.

Um aspecto relevante para a caracterização da cadeia cinemática dos veículos multi-articulados é a dimensão dos seus elementos. Elas definem as restrições de manobrabilidade da composição. A Figura 25 mostra as dimensões e as nomenclaturas relevantes, do ponto de vista de controle, da composição.

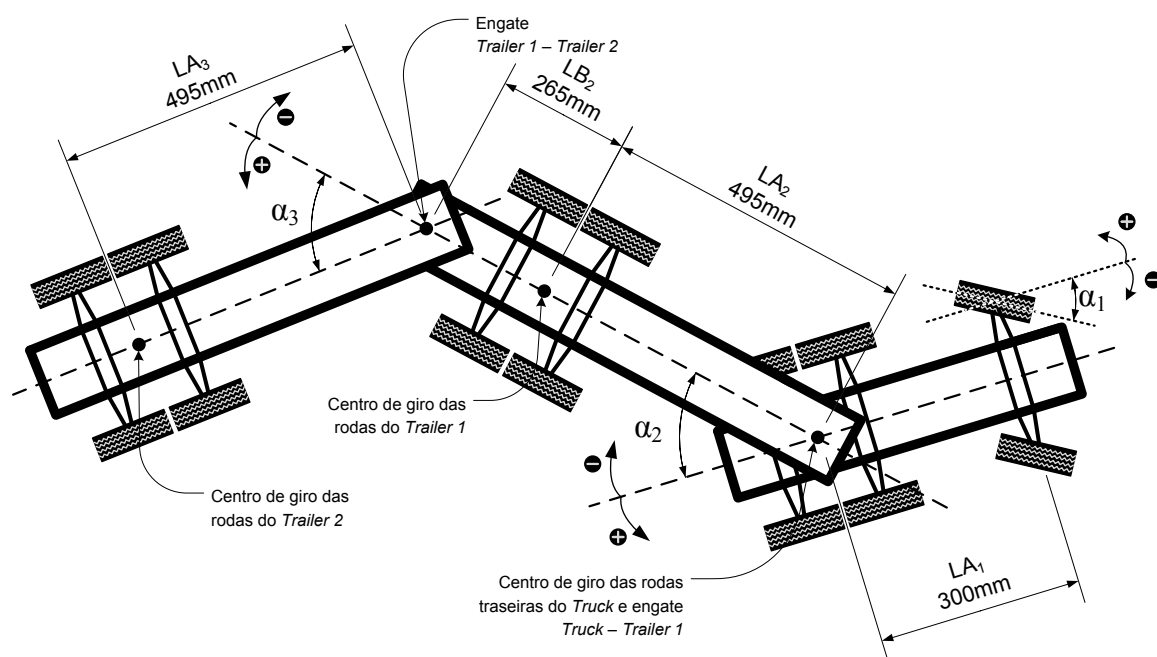


Figura 25: Definição dos ângulos e dimensões do protótipo.

Fonte: Adaptado de [1].

Nas seções a seguir, são apresentadas as principais contribuições de controle alcançadas por trabalhos anteriores, e se referem aos controladores CpPI e Global. Tais controladores servirão como base de desenvolvimento e comparação para o controlador proposto nessa dissertação.

3.2 Controlador CpPI

Em [3] foi proposta uma abordagem original para o controle de configuração de veículos multi-articulados em movimentos a ré. Essa proposta se caracteriza pela existência de um controlador *fuzzy* para cada articulação, formando uma arquitetura de controle em cascata.

O princípio de funcionamento dessa solução baseia-se na geração, por cada controlador, de uma referência para a articulação seguinte, ao final a ação de controle (α_1) é obtida.

Mais especificamente, o controlador de cada articulação recebe como única entrada, o erro a ser corrigido. Esse erro é obtido pela diferença entre o valor de referência, Ref_i , e o valor atual, α_i , do ângulo da articulação. Na sequência, o controlador infere o que deve ocorrer na próxima articulação e fornece como saída uma referência, Ref_{i+1} , para o controlador dessa articulação. Esse método é denominado CpPI (Controle por Propagação de Inferências). A Figura 26 mostra sua arquitetura quando aplicado a um veículo com três elementos.

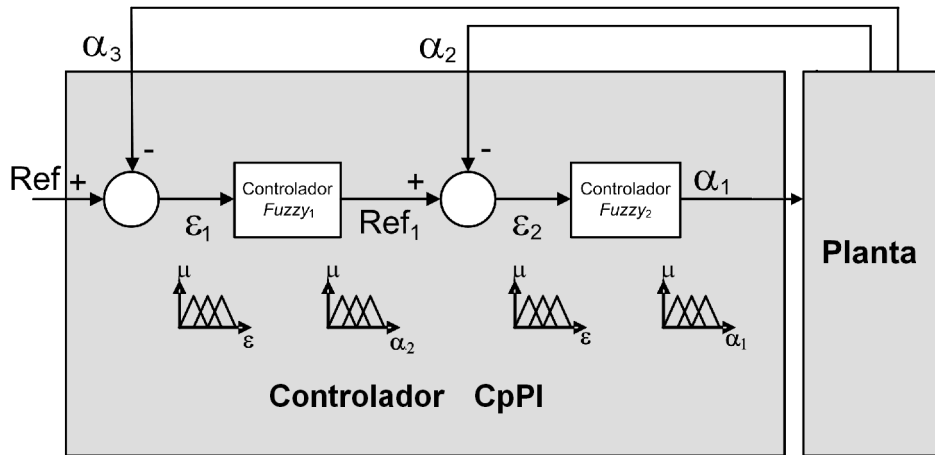


Figura 26: Controlador CpPI.

Uma das grandes vantagens do CpPI é a sua simplicidade de implementação e poder de generalização de controle para cadeias com múltiplos elementos. Cada controlador CpPI possui apenas uma entrada e uma saída, isso facilita a geração de regras e possibilita a existência de um banco de regras compacto, que conforme será visto mais a frente, favorece a velocidade de execução. Em contrapartida, sua arquitetura não permite que o controlador de uma dada articulação leve em consideração a configuração angular da articulação para a qual ele deve gerar a referência. Esse fato, como será visto, dificulta o

controle do veículo multi-articulado como um todo.

O controlador CpPI foi implementado e testado na plataforma experimental em [13], onde teve seus resultados apresentados. Objetivando-se melhorar os resultados apresentados pelo CpPI, uma nova topologia de controle foi proposta em [1] e será descrita na próxima seção.

3.3 Controlador Global

Foi desenvolvido em [1], um controlador baseado na premissa de que o controle do veículo fosse realizado por um único controlador *fuzzy*, dessa forma sendo denominado de Global.

O desenvolvimento do controlador Global, se deu inicialmente pelo uso da técnica de obtenção de regras *fuzzy* proposto por Wang-Mendel [45], que se baseia no processamento de dados coletados da planta em operação. Os dados utilizados foram extraídos de vários experimentos realizados no protótipo. Tais experimentos, buscavam o mapeamento das variações angulares das articulações do veículo em movimentos à ré. Após capturados pelo hardware embarcado, os dados foram processados em ferramentas de software específicas, formando assim uma base de dados disponível para análise. Tal base de dados, também foi útil para o desenvolvimento de uma solução de controle Neural, realizada em [14].

A técnica de Wang-Mendel obteve resultados insatisfatórios [46], gerando um grande número de regras inconsistentes. Dessa forma, o uso do conhecimento especialista se mostrou fundamental na geração da base de regras e consequentemente no desenvolvimento do controlador Global.

Aplicado ao protótipo em questão, o controlador Global é caracterizado por ser um sistema *fuzzy* que possui três entradas (α_2 , α_3 e *Erro*) e uma única saída (α_1), como mostra a Figura 27.

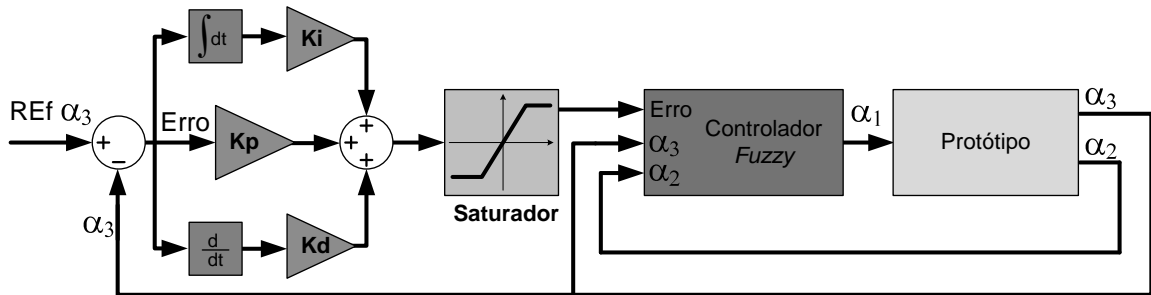
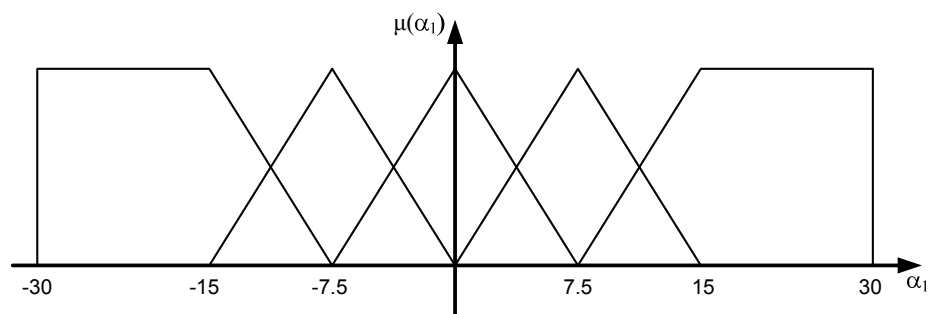
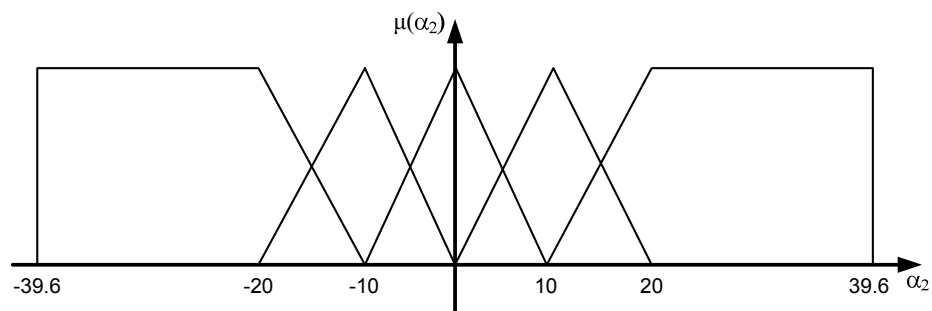


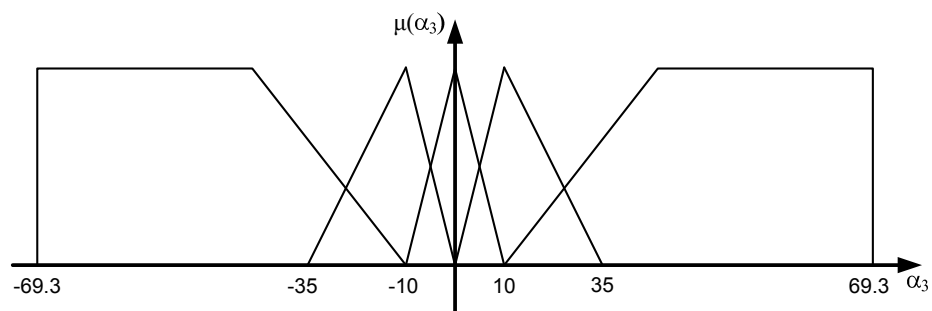
Figura 27: Controlador Global com PID.



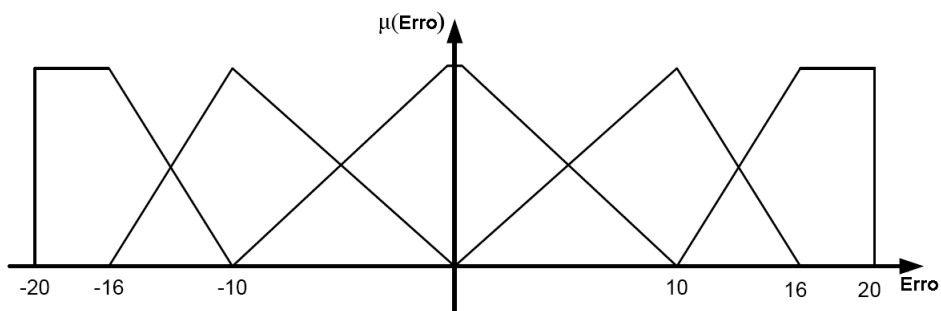
(a)



(b)



(c)



(d)

Figura 28: Formato das partições de todas as variáveis. Partições de α_1 em (a), α_2 em (b), α_3 em (c) e Erro em (d).

O controlador global realiza a leitura e o processamento simultâneo, no mesmo sistema *fuzzy*, de todos os ângulos da composição naquele instante. Assim, os ângulos das duas

articulações, α_3 e α_2 , são entradas de um único controlador.

Todas as variáveis presentes no controlador foram particionadas com granularidade igual a cinco termos. Os centros e as funções de pertinência de cada conjunto *fuzzy* foram selecionadas de maneira heurística, baseando-se nos valores máximos que cada uma delas pode assumir.

O banco de regras do controlador Global, possui 125 regras, posto que o sistema tem 3 entradas cada uma delas com 5 partições. Como dito anteriormente, as regras foram obtidas com base no conhecimento especialista adquirido em experimentações no protótipo real e numa sistematização do mapeamento causal entrada/saída. A Figura 28 exibe o particionamento das variáveis envolvidas. Para melhoria do desempenho do sistema de controle foi implementado um compensador PID com ganhos ajustados de maneira *ad-hoc* nos seguintes valores: $K_p = 1,5$, $K_d = 5$ e $K_i = 0,002$.

Por fim, nos experimentos realizados em [1] o controlador Global obteve vantagem de precisão em relação à versão do CpPI implementado [47], e terá seus resultados comparados, no Capítulo 4, com o controlador proposto a seguir.

3.4 Controlador CGPR

Ao contrário do conceito de controle global, onde todas entradas são avaliadas em um único controlador, o controlador aqui proposto, denominado Controle no Giro por Propagação de Referência (CGPR), utiliza o conceito de cascadeamento de controladores empregado no CpPI e, assim, mantendo a simplicidade e o poder de generalização presentes nesse controlador.

A abordagem em cascata, permite o uso de um controlador para cada semi-reboque. Isso facilita a expansão do número de elementos da composição, pois possibilita a reutilização da mesma estrutura do controlador, obviamente configurada de acordo com as características de cada elemento. Deve-se ressaltar que cada controlador deve operar satisfatoriamente sob pena de comprometer o controle do veículo como um todo.

Uma das principais inovações do CGPR, provém da definição de ângulo crítico e da formulação para sua obtenção, obtidos por Kulitz [3]. Os ângulos críticos definem a faixa de variação permissível para o ângulo entre duas articulações consecutivas. Para ângulos acima desse valor crítico não existe qualquer ação de controle que reverta a condição de *jackknife* ou engavetamento.

Contudo, a recíproca não é verdadeira, ou seja, mesmo tendo-se uma configuração na qual os ângulos das articulações são menores do que o valor crítico, isso não garante que seja possível controlar a composição. Dessa forma, independentemente do tipo de controle que esteja sendo utilizado o engavetamento pode ser inevitável. Nesse caso, é dito que a composição estava inicialmente em uma configuração crítica, que será melhor explicada ainda nesse seção.

A formulação para obtenção dos ângulos críticos se baseia na condição em que o centro de giro dos elementos da composição coincidem ao se impor o máximo ângulo permitido na direção. Nessa situação, o movimento é circular e possui o menor raio de curvatura possível.

Conhecendo-se o máximo ângulo permissível das rodas de direção, determina-se o ângulo crítico entre o elemento trator e o primeiro semi-reboque. Na sequência, pode-se calcular o ângulo crítico entre o primeiro e o segundo semi-reboque e assim sucessivamente até o cálculo do ângulo crítico da última articulação. Conforme em Kulitz [3] as equações que definem o ângulo crítico entre dois elementos consecutivos são:

$$\alpha_{n+1_{max}} = \text{sen}^{-1} \left(\frac{LA_{n+1} \times \cos(\gamma)}{\beta} \right) + \gamma \quad (3.1)$$

$$\gamma = \tan^{-1} \left(\frac{LB_n}{\beta} \right) \quad (3.2)$$

$$\beta = \left(\frac{LA_n \times \cos(\alpha_n) + LB_n}{\text{sen}(\alpha_n)} \right) \quad (3.3)$$

As constantes LA_n , LB_n se referem as dimensões de cada elemento da composição, conforme mostrado na Figura 25. Para o protótipo os valores dos ângulos críticos das articulações foram determinados utilizando-se as dimensões da Figura 25 e $\alpha_{1_{max}} = \pm 20^\circ$, sendo esse valor decorrente da limitação mecânica do sistema de direção. Os valores obtidos foram: $\alpha_{2_{critico}} = \pm 36.9^\circ$ e $\alpha_{3_{critico}} = \pm 66.2^\circ$.

Sabendo-se que essa formulação foi obtida numa situação em que o movimento da composição é circular, pode-se fazer uso dessas mesmas equações, não apenas para o cálculo dos ângulos críticos, mas também, para auxiliar o controle da composição.

Dado uma referência para o ângulo da última articulação, *Ref* α_n , pode-se, de modo recursivo, através da inversa das equações acima, determinar os valores de giro α_{n-1}^* , α_{n-2}^* , e assim sucessivamente, até α_1^* que manterão a referência *Ref* α_n constante (composição

em giro). Essa configuração final pode ser representada pelo vetor $[Ref \alpha_n, \alpha_{n-1}^*, \alpha_{n-2}^*, \dots, \alpha_1^*]$.

Assim, rearranjando as Equações 3.1, 3.2 e 3.3 de forma a se evidenciar o termo α_n , chega-se na Equação 3.4, utilizada para o cálculo do ângulo de giro de dois elementos consecutivos.

$$\alpha_{n_{giro}} = \tan^{-1} \left(\frac{LA_n \times \sin(\alpha_{n+1})}{LA_{n+1} + LB_n \times \cos(\alpha_{n+1})} \right) \quad (3.4)$$

Contudo, a Equação 3.4 não informa como atingir a configuração de referência $[Ref \alpha_n, \alpha_{n-1}^*, \alpha_{n-2}^*, \dots, \alpha_1^*]$, a partir da configuração atual, dada por $[\alpha_n, \alpha_{n-1}, \alpha_{n-2}, \dots, \alpha_1]$. Como será visto, a convergência para esses valores deve ser realizada segundo um critério que favoreça o controle do sistema no momento em que a referência é alcançada. Esse critério é a simultaneidade em que os ângulos das articulações devem convergir para os seus respectivos valores de referência.

Dado a configuração atual do veículo, $[\alpha_n, \alpha_{n-1}, \alpha_{n-2}, \dots, \alpha_1]$ e uma $Ref \alpha_n$ a ser atingida, o CGPR calcula inicialmente o ângulo, $\alpha_{n-1_{giro}}$, para a $(n-1)$ -ésima articulação que mantém a articulação α_n em giro. Entretanto, para fazer com que α_n tenda para $Ref \alpha_n$ é adicionado uma parcela, h_{n-1} , ao $\alpha_{n-1_{giro}}$ calculado, fornecendo-se, assim, a referência $Ref \alpha_{n-1} = h_{n-1} + \alpha_{n-1_{giro}}$, a ser seguida pela $(n-1)$ -ésima articulação. O mesmo cálculo é realizada para as articulações seguintes, obtendo-se ao final o valor da variável de controle α_1 . A repetição continuada desse procedimento, conduz a composição, inicialmente na configuração $[\alpha_n, \alpha_{n-1}, \alpha_{n-2}, \dots, \alpha_1]$, para a configuração final $[Ref \alpha_n, \alpha_{n-1}^*, \alpha_{n-2}^*, \dots, \alpha_1^*]$ através da geração de referências intermediárias para os ângulos das articulações.

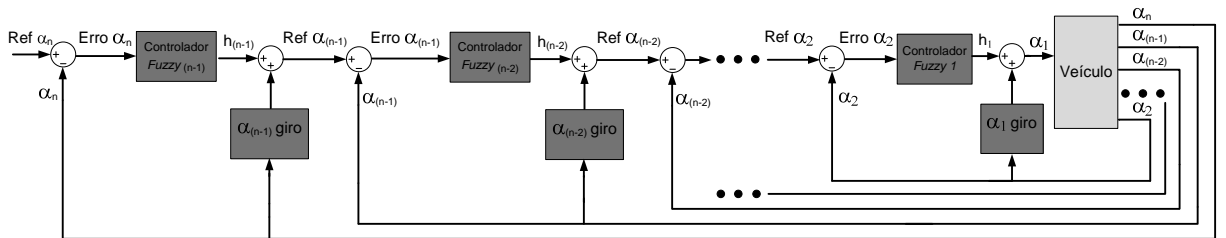


Figura 29: Arquitetura do CGPR inicialmente proposta.

Conforme mostra a Figura 29, os valores $h_{n-1}, h_{n-2}, \dots, h_1$ são obtidos por um sistema fuzzy que possui como entrada apenas o erro dos ângulos das articulações.

O módulo de h define as características dinâmicas do controlador. Valores elevados implicam em maior velocidade de resposta, necessárias em determinadas manobras. Contudo, em outras situações, esse mesmo valor de h , pode fazer com que a composição não atinja a referência com posicionamento apropriado, ou seja, sem os ângulos das demais articulações próximos a suas respectivas referências.

Neste momento, cabe ressaltar que a determinação da quantidade h está relacionada ao conhecimento especialista e, portanto, o emprego da teoria *fuzzy* justifica-se devido a sua facilidade de representar este conhecimento em termos matemáticos.

Nos experimentos realizados na plataforma experimental, o sistema *fuzzy* utilizado é do tipo relacional ou Mandani [48] e foram escolhidos cinco termos lingüísticos para cada variável, sendo eles, *NG* (Negativo Grande), *NP* (Negativo Pequeno), *ZE* (Zero), *PP* (Positivo Pequeno) e *PG* (Positivo Grande).

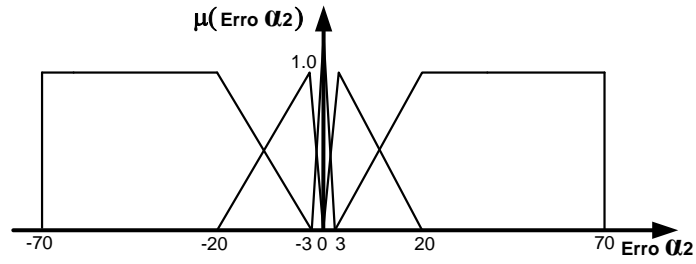
Por simplicidade, as funções de pertinência utilizadas são do tipo triangular e o universo de discurso das variáveis foram baseados nos ângulos críticos das articulação. A Figura 30, mostra as partições das variáveis para os controladores de α_2 e α_3 . Diferentemente das funções, ambos controladores utilizaram o mesmo banco de regras, exibido na Tabela 4.

Tabela 4: Base de regras dos controladores.

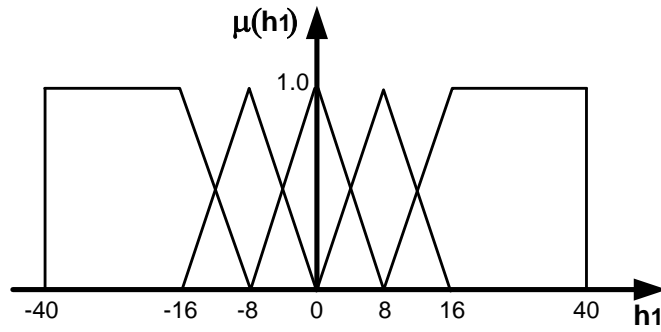
Regra	Entrada $Erro \alpha_n$	Saída h_{n-1}
1	NG	NG
2	NP	NP
3	ZE	ZE
4	PP	PP
5	PG	PG

Essa versão inicial do CGPR apresentou melhorias em relação aos controladores CpPI e Global. Contudo, ainda não foi possível atingir valores de referência maiores que 30° para α_3 . Além disso, como acontecia nos controladores anteriormente propostos, nos testes em que se pretendia alinhar α_3 partindo-se de um ângulo diferente de zero, a composição entrava em *jackknife*.

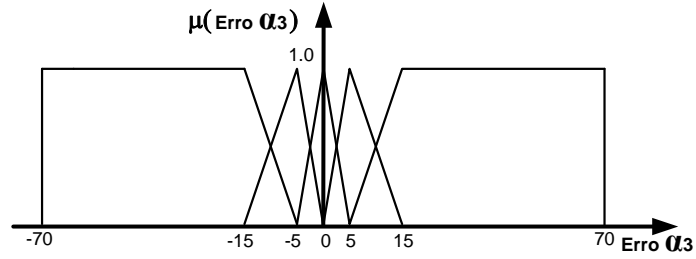
Tais limitações estão relacionadas ao fato de o controlador não impedir que a composição atingisse uma configuração crítica. Como mencionado, uma configuração crítica é aquela em que mesmo tendo-se os ângulos das articulações menores do que os seus respectivos ângulos críticos a composição tende invariavelmente para o engavetamento.



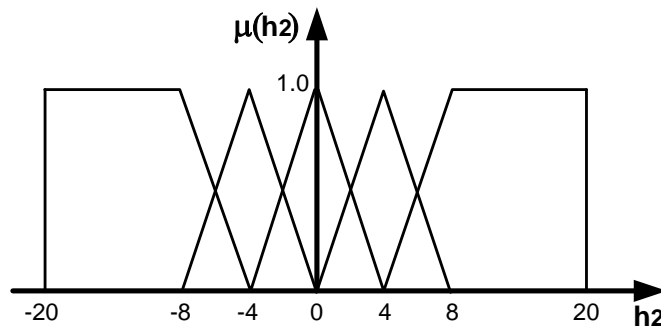
(a)



(b)



(c)



(d)

Figura 30: Particionamento das variáveis. Em (a) $Erro \alpha_2$. Em (b) h_1 . Em (c) $Erro \alpha_3$. Em (d) h_2 .

Isso acontece quando a distância necessária a ser percorrida para trás, para que α_2 atinja o valor $\alpha_{2_{giro}}$, que cessará a variação de α_3 , é superior à distância que levará α_3 ao

seu valor crítico. Esse retardo de α_2 em atingir o ângulo de controle de α_3 é referenciado como "atraso" da resposta do sistema. O mesmo raciocínio se aplica para quaisquer outros dois elementos consecutivos de uma composição com n elementos.

Obviamente, esse atraso varia de acordo com a configuração da composição. Quanto maior for o ângulo de uma articulação, ou seja, quanto mais próximo ele estiver do seu valor crítico, menor será a taxa de variação desse ângulo por unidade de deslocamento que se conseguirá obter no movimento de fechamento desse ângulo, que é necessário para se evitar o engavetamento. Ao contrário, quanto maior o ângulo da articulação maior é a taxa de variação que se pode obter no sentido de aumentar o módulo desse ângulo (movimento de abertura), o que favorece o alcance do ângulo pela articulação.

No caso específico da aplicação no protótipo, isso ocorre porque quanto mais próximo α_2 está do seu valor crítico, menor é a margem do ângulo de controle, α_1 , para a realização do movimento de fechamento, em contrapartida, maior é margem para o movimento de abertura. A Figura 31 exemplifica o exposto. Essa "lentidão" no fechamento de α_2 , contribui para o aumento do atraso de resposta, podendo não ser tolerado para o controle de α_3 .

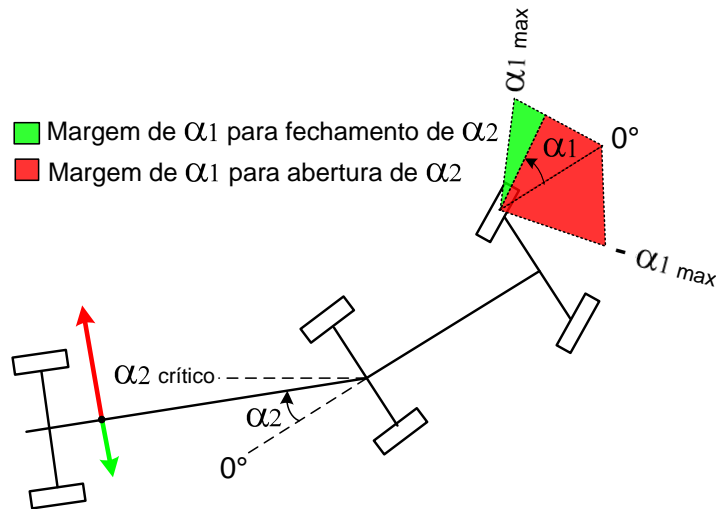


Figura 31: Variação da velocidade de abertura e fechamento de α_2 .

Para diminuir o atraso nessa região de resposta mais lenta, pode-se diminuir a diferença entre os valores de α_2 e $\alpha_{2_{giro}}$. Isso é obtido gerando-se referências para α_2 próximas ao valor $\alpha_{2_{giro}}$, através da diminuição do módulo de h_2 , uma vez que, $Ref \alpha_2 = h_2 + \alpha_{2_{giro}}$.

Para tanto, foi adicionada a variável α_2 como entrada no sistema *fuzzy* de controle de α_3 . Essa medida evita que referências intermediárias geradas pelo controlador de α_3

enviem α_2 rapidamente para a região de resposta lenta ao movimento de fechamento. Quando aplicado ao protótipo o CGPR possui estrutura mostrada na Figura 32. A forma geral da arquitetura do controlador CGPR que considera o valor do ângulo da articulação seguinte é mostrado na Figura 33.

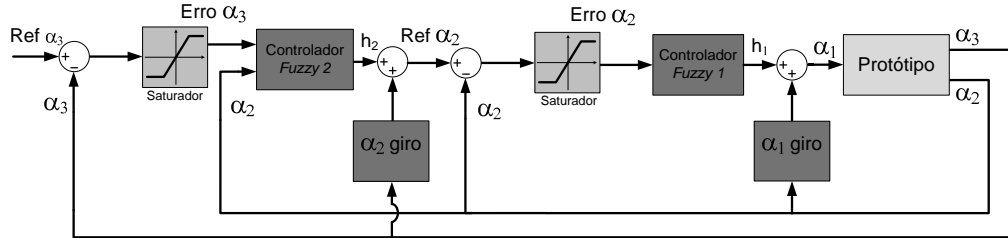


Figura 32: Arquitetura do controlador CGPR para duas articulações.

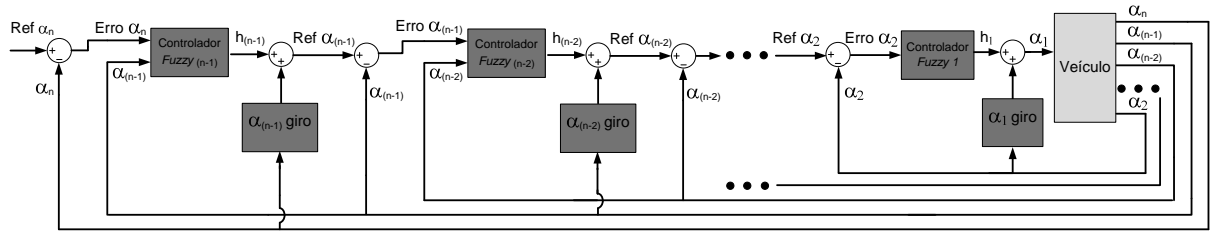


Figura 33: Estrutura geral do CGPR.

Como α_1 é a variável de controle, o controlador do ângulo α_2 não recebe α_1 como entrada, sendo mantida para o controlador dessa articulação as mesmas partições e regras definidas anteriormente. A variável α_2 , incluída no controlador de α_3 , também possuiu funções de pertinência triangulares e cinco termos lingüísticos. Assim, o número de regras desse controlador aumentou de 5 para 25. A Figura 34 mostra o particionamento da nova variável de entrada α_2 .

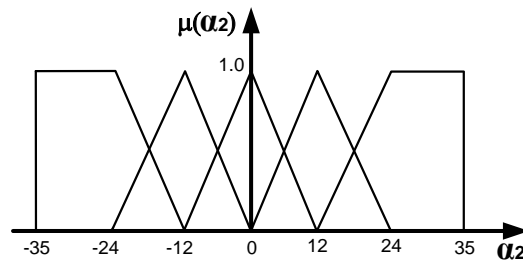


Figura 34: Particionamento de α_2 .

As Tabela 5 mostra as novas regras definidas com a inclusão da entrada α_2 . As regras 1, 5, 21 e 25 mostram saídas pequenas para h_2 , quando α_2 é grande.

Tabela 5: Base de regras do controlador de α_3 .

Regra	Entrada α_2	Entrada <i>Erro</i> α_3	Saída h_2
1	NG	NG	NP
2	NG	NP	NP
3	NG	ZE	ZE
4	NG	PP	PP
5	NG	PG	PP
6	NP	NG	NG
7	NP	NP	NP
8	NP	ZE	ZE
9	NP	PP	PP
10	NP	PG	PG
11	ZE	NG	NG
12	ZE	NP	NP
13	ZE	ZE	ZE
14	ZE	PP	PP
15	ZE	PG	PG
16	PP	NG	NG
17	PP	NP	NP
18	PP	ZE	ZE
19	PP	PP	PP
20	PP	PG	PG
21	PG	NG	NP
22	PG	NP	NP
23	PG	ZE	ZE
24	PG	PP	PP
25	PG	PG	PP

A Figura 35 apresenta um recurso muito interessante para a análise de sistemas *fuzzy*, que é a superfície formada pelas variáveis de entrada e saída. Observa-se na Figura 35 que o valor de h_2 varia de acordo com o erro e o valor de α_2 , apresentando maior módulo na região central e sendo saturado nas extremidades.

A redução do valor de h_2 , introduzida para evitar que a composição atinja uma configuração crítica em determinadas situações, traz um efeito indesejado que é a diminuição da capacidade de eliminação do erro em regime. Isso acontece porque a redução do módulo de h_2 possui um efeito de diminuição do ganho do controlador. Para a atenuação desse efeito, foram introduzidos compensadores do tipo PID, conforme será visto no Capítulo 4.

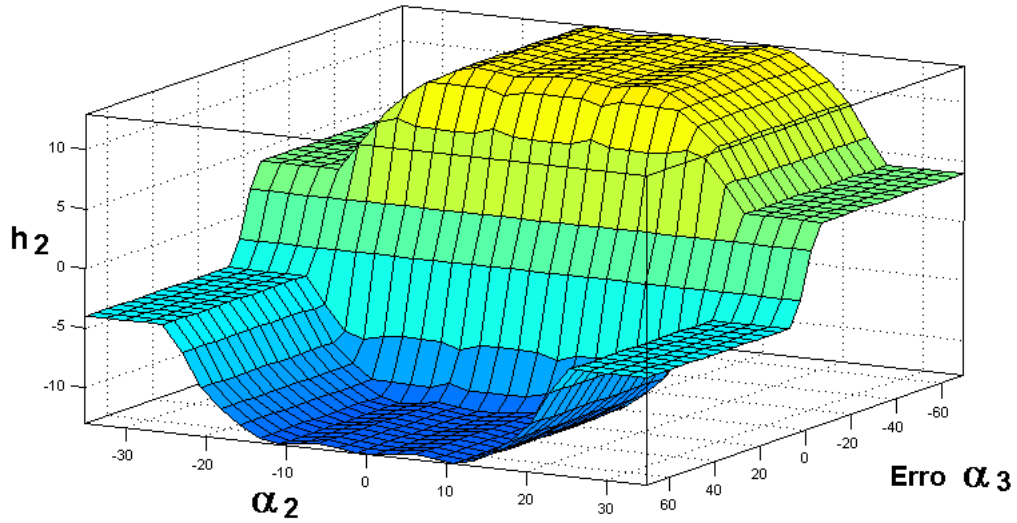


Figura 35: Superfície de decisão para o controlador *fuzzy*

3.5 Esforço computacional

Um aspecto relevante sobre os controladores mencionados, é o esforço computacional por eles requerido. Em aplicações de tempo real, o desempenho alcançado pelo controlador é diretamente afetado pelo tempo de execução do seu algoritmo. Em sistemas *fuzzy*, o número de regras presentes no banco de regras é um indicativo desse esforço computacional. As Equações 3.5, 3.6 e 3.7 determinam, em função do número de articulações (n) e da granularidade das partições (p), o número total de regras (R) geradas para os controladores CpPI, CGPR e Global respectivamente.

$$R(n, p) = n \times p \quad (3.5)$$

$$R(n, p) = (n - 1) \times p^2 + p \quad (3.6)$$

$$R(n, p) = p^{(n+1)} \quad (3.7)$$

Conforme exemplificado na Tabela 6, para um dado valor de n e p , o controlador CpPI apresenta o menor número de regras em relação aos demais. O CGPR está na posição intermediária, apresentando um número de regras ligeiramente superior ao do CpPI. O controlador Global apresentou o maior número de regras, isso se deve à característica puramente exponencial da equação 3.7.

Tabela 6: Número de regras geradas por cada controlador em função do número de articulações e da quantidade de termos empregados no particionamento das variáveis de entrada.

N. Articulações (n)	$n = 2$		$n = 3$	
N. Termos (p)	$p = 5$	$p = 7$	$p = 5$	$p = 7$
$R(n, p)$ CPpI	10	14	15	21
$R(n, p)$ CGPR	30	56	55	105
$R(n, p)$ Global	125	343	625	2401

3.6 Conclusão

Este capítulo apresentou diferentes abordagens de controle *fuzzy* para a solução do problema de controle de configuração em movimentos à ré de veículos multi-articulados. Foram descritas as principais contribuições de trabalhos anteriores, que serviram como base e comparação para a nova abordagem de controle apresentada.

A obtenção da Equação 3.4, que define o ângulo de giro para um par de articulação, permitiu a determinação da configuração final da composição para uma dada referência da última articulação, essa configuração é representada pelo vetor $[Ref \ \alpha_n, \alpha_{n-1}^*, \alpha_{n-2}^*, \dots, \alpha_1^*]$. Além disso, essa equação permitiu que o CGPR da n -ésima articulação atuasse na região ao entorno do seu ângulo de giro atual, contribuindo com a geração mais eficiente de referências para a $(n-1)$ -ésima articulação.

O desenvolvimento do CGPR levou em consideração a configuração crítica, situação na qual a composição tende ao engavetamento mesmo tendo os ângulos das suas articulações menores que os respectivos valores críticos. Outra contribuição fornecida, foi a caracterização do atraso dinâmico inerente a sistemas dessa natureza. Tais atrasos, quando suficientemente grandes, não permitem que a $(n-1)$ -ésima articulação atinja o ângulo necessário para controlar a n -ésima articulação, a tempo de evitar que essa última ultrapasse seu valor crítico, portanto, o atraso esta diretamente relacionado à condição de configuração crítica.

A compreensão do efeito do atraso sobre o controle, permitiu a melhoria do CGPR, ao se evitar que o controlador gerasse referências que levassem a composição a uma configuração crítica. Essa melhoria foi possível por meio da realimentação no controlador da n -ésima articulação com o ângulo da $(n-1)$ -ésima articulação. Ao final, obteve-se uma

arquitetura geral do CGPR para um sistema de com n articulações.

Por fim, uma versão do CGPR com o uso de compensadores do tipo PID foi desenvolvida para a validação prática desse controlador no protótipo desenvolvido. Assim, o próximo capítulo se dedica a mostrar resultados experimentais do CGPR, sendo estes comparados aos obtidos pelo Controlador Global.

4 *Resultados Experimentais*

4.1 Introdução

Objetivando-se validar, analisar e otimizar o desempenho do controlador proposto, foram realizados diversos experimentos na plataforma experimental desenvolvida. A Figura 36 mostra os principais recursos utilizados nos experimentos práticos. O computador remoto, onde são realizados os cálculos de controle, possui processador Intel Centrino com frequência de 1.6Ghz e 2GB de memória RAM.

Inicialmente foi utilizado um *joystick* para geração de referências para o ângulo α_3 , porém, a comunicação com esse dispositivo alocou um parcela considerável do processamento, interferindo negativamente no desempenho geral, portanto, não sendo utilizado nos experimentos. Assim, as referências eram inseridas diretamente no algoritmo de controle.

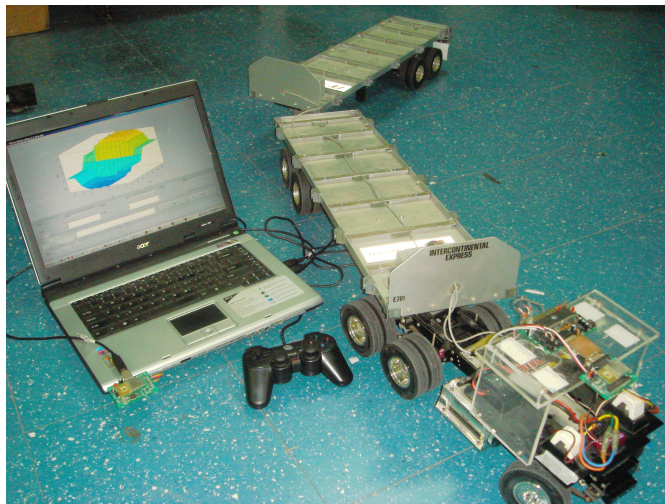


Figura 36: Ferramentas utilizadas nos experimentos

O sistema computacional utilizado, permitiu que a variável de controle fosse atualizada no protótipo a cada 50ms aproximadamente, sendo a maior parte desse tempo gasto na comunicação de dados(40ms). A quantificação desses tempos foi possível através do uso

das funções *tic* e *tac* do matlab. As manobras foram realizadas numa velocidade média de 10cm/s , o que proporcionalmente ao veículo em escala real corresponde a 5km/h , sendo portanto, uma velocidade plausível para manobras à ré. É possível a realização de experimentos com velocidades superiores, porém, as custas da diminuição do número de atualizações do controle por distância percorrida. Em uma aplicação final, o tempo de atualização da ação de controle na planta, pode ser sensivelmente reduzido com execução dos algoritmos de controle em um computador embarcado no veículo, como realizado em [16], onde a velocidade em escala de 70Km/h foi alcançada pelo veículo multi-articulado.

O algoritmo de controle foi implementado utilizando-se o *toolbox fuzzy* do Matlab[®] [49]. Essa ferramenta é muito útil na implementação de sistemas *fuzzy*, pois permite a fácil manipulação das regras e do particionamento das variáveis envolvidas. O Anexo C contém o *script* desenvolvido que implementa o controlador CGPR.

O experimentos foram divididos em duas partes, a primeira parte, Seção 4.2, se refere aos experimentos de ajuste dos compensadores utilizados nas malhas de realimentação do CGPR para a primeira e segunda articulações. A segunda parte, Seção 4.3, exhibe experimentos de controle de configuração do controlador CGPR e faz uma comparação com os resultados alcançados pelo controlador Global em [1].

4.2 Ajustes em Malha Fechada do CGPR

Para um resultado satisfatório do controle de configuração do veículo é imprescindível que o controlador de cada articulação tenha um bom desempenho individualmente. Isso é necessário porque o sinal de referência no CGPR é propagado em cascata e, portanto, um único controlador mal ajustado pode gerar referências erradas para as demais articulações, induzindo ações incorretas de controle ao final do processo.

O ajuste de cada controlador foi realizado de forma sequencial, ou seja, após o controlador da primeira articulação ter sido ajustado, o segundo *trailer* foi acrescentado e o seu controlador também ajustado. Caso existisse mais elementos na composição, esse procedimento seria sucessivamente repetido.

Além dos ajustes nas regras e no particionamento das variáveis *fuzzy*, que levam em consideração o universo de discurso das variáveis e o conhecimento especialista, foram implementados compensadores do tipo PID. Esses compensadores foram utilizados na malha de realimentação e possuem a função de diminuir o erro em regime, melhorar a velocidade de resposta, diminuir possíveis oscilações e fornecer maior robustez ao sistema contra

distúrbios externos. Os ganhos desses controladores foram determinados de maneira *ad hoc* por meio da análise de diversos experimentos.

4.2.1 Controlador da Primeira Articulação: *Truck-Trailer*

Para o ajuste dos compensadores do controlador da primeira articulação (α_2) foram utilizados apenas dois elementos na composição, ou seja, o elemento trator e um elemento passivo. Essa articulação especificamente, se difere das demais pelo fato de ser controlada diretamente pelo ângulo das rodas de direção, que pode ser instantaneamente modificado pelo servo motor. Essa característica elimina o atraso de configuração para essa articulação, tornando o seu controle mais rápido e mais preciso que as demais.

Apenas com o compensador proporcional, que é inerente a própria realimentação com o erro de α_2 , o controlador já apresentava bons resultados para referências constantes e em degrau, conforme mostra as Figuras 37(a) e 38(a). Contudo, para referências em rampa, Figura 39(a), o controle apresentou um erro em regime. O experimento com referência em rampa é muito importante de ser avaliado, pois, quando a segunda articulação é inserida, o seu controlador irá gerar uma referência com essa característica para o ângulo da primeira articulação.

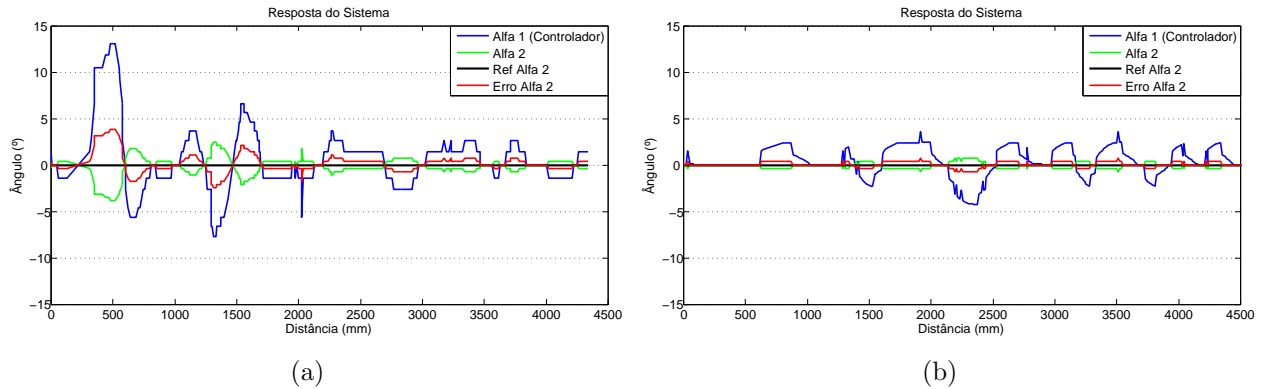


Figura 37: $Ref_{\alpha_2} = 0^\circ$ (a) Compensador proporcional (b) Compensador proporcional e integral

Para reduzir o erro em regime, sobretudo para referências em rampa, foi acrescentado o termo integrador, que conforme pode ser visto na Figura 39(b) diminuiu esse erro. O termo integrador, dado por $\Sigma(Erro_{\alpha_2}) * Ki$ sendo Ki uma constante, foi implementado através de uma janela móvel de 10 amostras do erro de α_2 , ou seja, a cada nova amostra que era somada a mais antiga era descartada do somatório. Os ganhos do compensador PI foram fixados em $Kp = 1$ e $Ki = 1/10$.

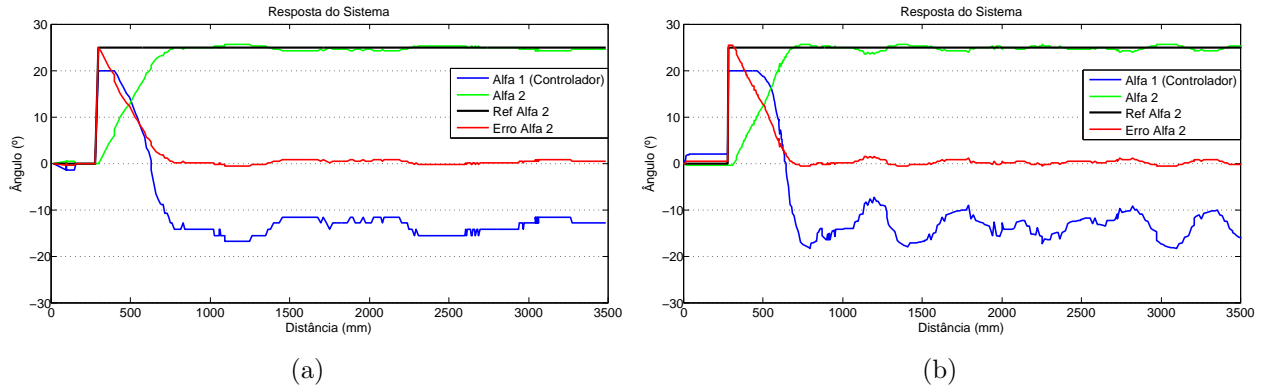


Figura 38: $Ref_{\alpha_2} = 25^\circ$ (a) Compensador Proporcional (b) Compensador Proporcional e Integral

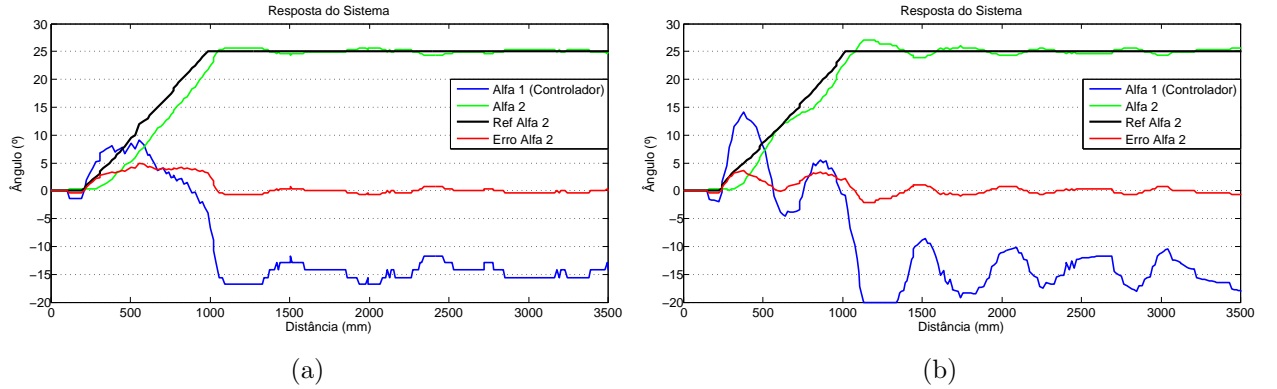


Figura 39: Referência em rampa para α_2 (a) Compensador Proporcional (b) Compensador Proporcional e Integral

4.2.2 Controlador da Segunda Articulação: *Truck-Trailer-Trailer*

Conforme mencionado no Capítulo 3, o sistema *fuzzy* foi projetado para que a variável de saída h_2 , assumia diferentes valores de acordo com a configuração da veículo multi-articulado. Devido a essa variação do valor de h_2 , que funciona como um ganho do controlador, ocorrem situações em que esse valor é insuficiente para eliminar o erro em regime, conforme pode ser visto na Figura 40(a), onde foi dado uma referência de -20° para α_3 .

Para suprir a deficiência do controlador nessas situações foi adicionado um compensador integrador. A Figura 40(b) mostra que a presença do termo integrador eliminou o erro em regime. Porém, a resposta ficou mais oscilatória, sobretudo nas configurações em que o ganho, h_2 , já era suficientemente grande e, assim, o termo integrador acelerava ainda mais a tendência do movimento através da geração de referências mais acentuadas para o controlador de α_2 .

O efeito da sobreposição dos ganhos devido a h_2 e o termo integrador, fica evidenciado observando-se o comportamento de α_2 nos experimentos da Figura 41 em que α_3 partiu de 20° e foi dada a referência $Ref\alpha_3 = 0$. A Figura 41(a) mostra o resultado alcançado com o uso do termo proporcional apenas. Na Figura 41(b) pode-se observar o aumento da oscilação da resposta, devido ao efeito da inclusão do termo integrador.

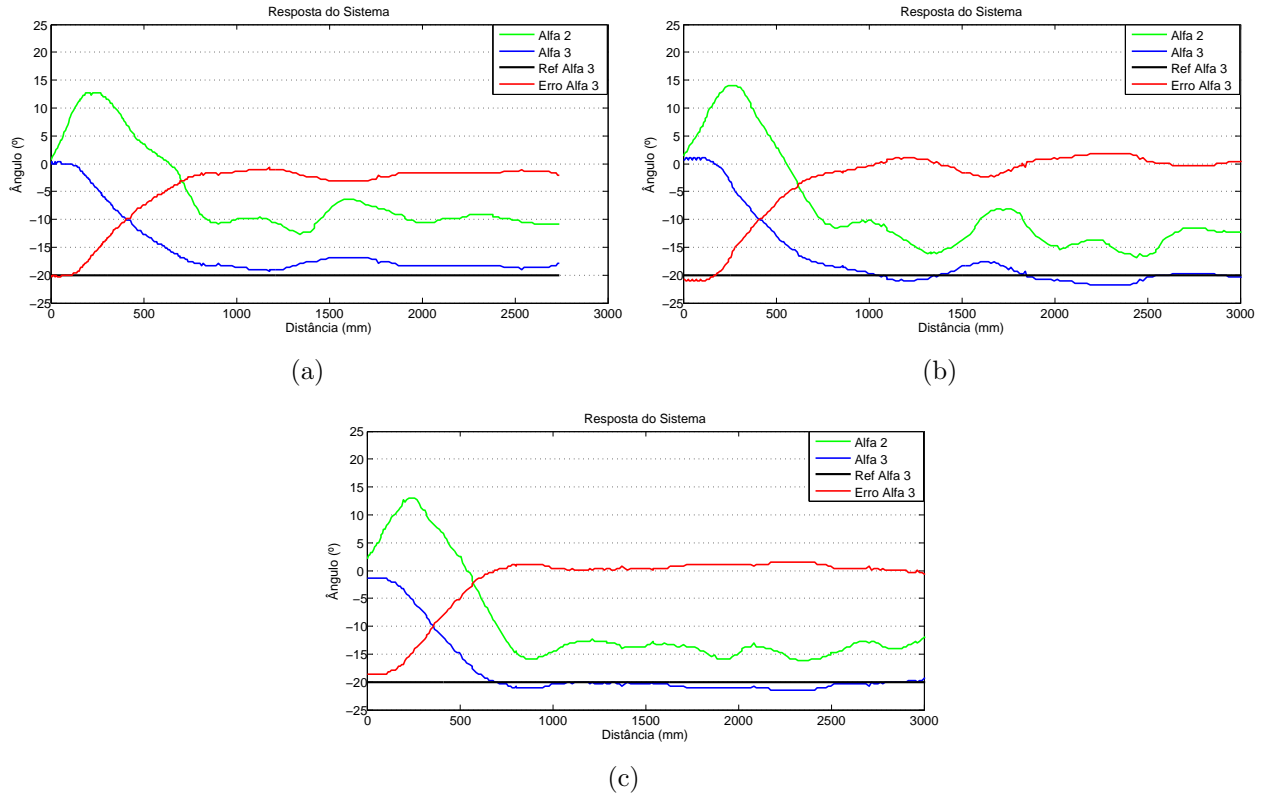


Figura 40: (a) Compensador proporcional (b) Compensador proporcional e integral (c) Compensador proporcional, Integral e derivativo

A oscilação e o *overshoot* de α_2 , favorecem a ocorrência do engavetamento da composição. Afim de se reduzir esses efeitos indesejáveis, foi inserido o termo derivativo no compensador de realimentação. Isso proporcionou geração de referências mais suaves para o controlador de α_2 nas situações em que α_3 já tendia para a referência, conforme pode ser visto nas Figuras 40(c) e 41(c).

O termo derivativo é dado pela expressão $Kd * (Erro\ \alpha_{3atual} - Erro\ \alpha_{3anterior})$, onde Kd é uma constante. Para o controlador de α_3 as constantes do compensador foram definidas em $Kp = 1$, $Ki = 1/40$ e $Kd = 1/2$. A Figura 42 mostra o esquemático do controlador CGPR implementado com a presença dos compensadores.

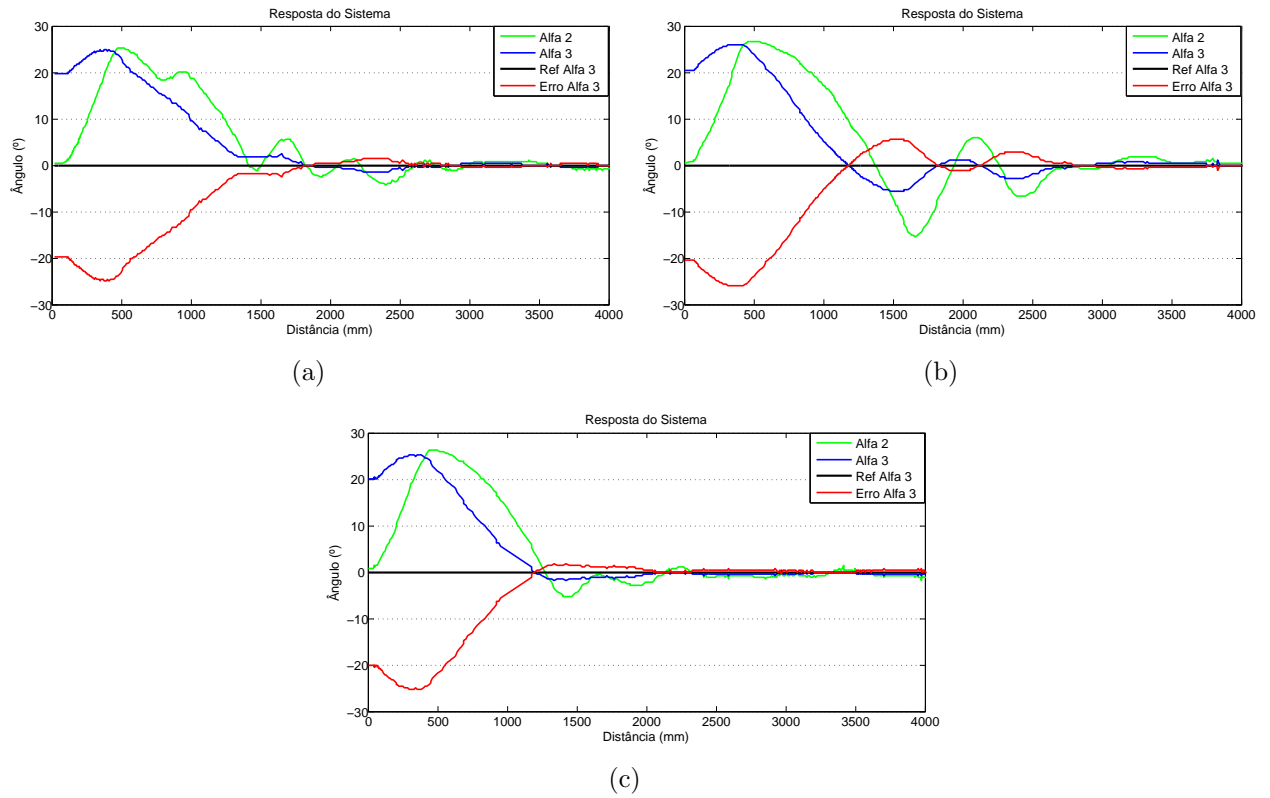


Figura 41: (a) Compensador proporcional (b)Compensador proporcional e integral (c) Compensador proporcional, Integral e derivativo

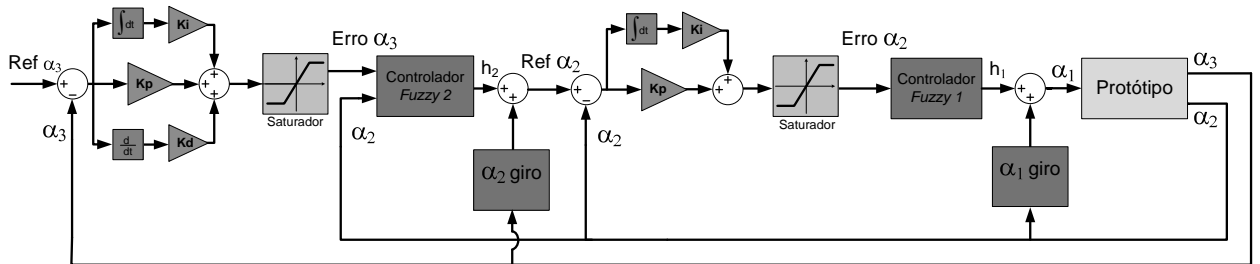


Figura 42: Arquitetura final do controlador CGPR.

4.3 Experimentos: Controlador Global Vs CGPR

Esta seção tem por objetivo apresentar os resultados de controle de configuração obtidos pelo controlador CGPR. Além disso, esses resultados são comparados aos obtidos pelo controlador Global. Os experimentos foram realizados para diferentes configurações iniciais e referências para α_3 , sendo que cada experimento é identificado pelo vetor $[Ref \alpha_3, \alpha_3, \alpha_2]$, que representa a referência requerida e a condição inicial da composição.

4.3.1 Experimentos I

Nesse experimento, caracterizado por $[0^\circ, \approx 0^\circ, \approx 0^\circ]$, o veículo partiu aproximadamente alinhando e foi requerido que essa configuração fosse mantida.

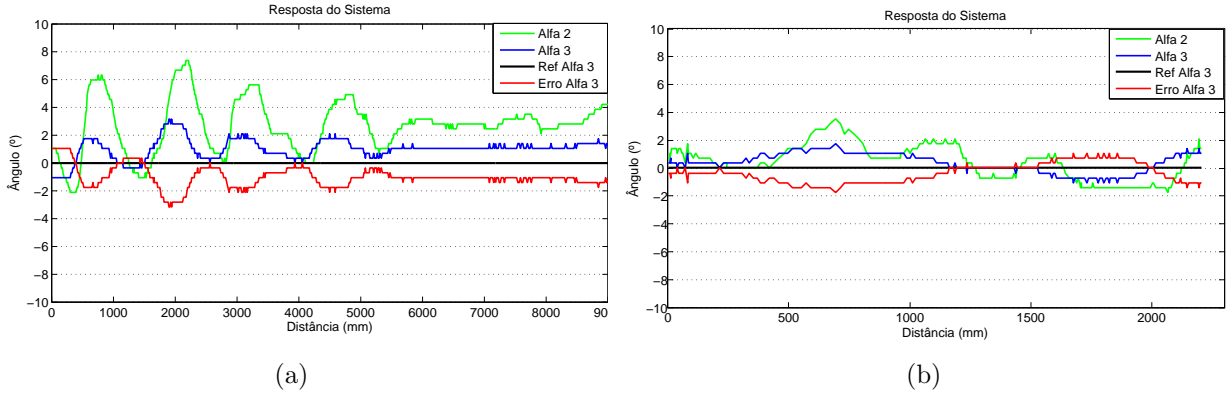


Figura 43: $Ref \alpha_3 = 0^\circ$ (a) Controlador Global (b) Controlado CGPR

Como pode ser visto na Figura 43, ambos controladores mantiveram o ângulo a ser controlado, α_3 , próximo a referência com precisão de aproximadamente 2° . Essa precisão é considerada satisfatória levando-se em consideração todos os erros envolvidos na discretização das grandezas e nas limitações mecânicas do sistema. Desse experimento observa-se também que a resposta obtida pelo controlador CGPR é menos oscilatória.

4.3.2 Experimentos II

No segundo experimento foi requisitada uma referência nula para α_3 partindo-se com $\alpha_2 \approx 0$ e $\alpha_3 \neq 0$. O grau de dificuldade desse manobra aumenta com o aumento do ângulo inicial de α_3 .

Conforme pode ser visto nos gráficos da Figura 44, o ângulo α_3 cresce, divergindo da referência, até que α_2 atinja o valor que cesse essa tendência. A partir desse ponto, é fundamental controlar a velocidade de convergência de α_3 para a referência, isso permite a α_2 , que se encontra em uma região de fechamento lento, atingir o valor necessário que manterá α_3 em um ponto próximo da referência.

O controle da velocidade com que α_3 converge é realizado gerando-se referências intermediárias para α_2 próximas ao valor de giro de α_3 , isso é conseguido através da diminuição do módulo de h_2 , uma vez que $Ref \alpha_{n-1} = h_{n-1} + \alpha_{n-1_{giro}}$.

A Figura 44(a) mostra o experimento, caracterizado por $[0^\circ, 15^\circ, \approx 0^\circ]$, realizado com

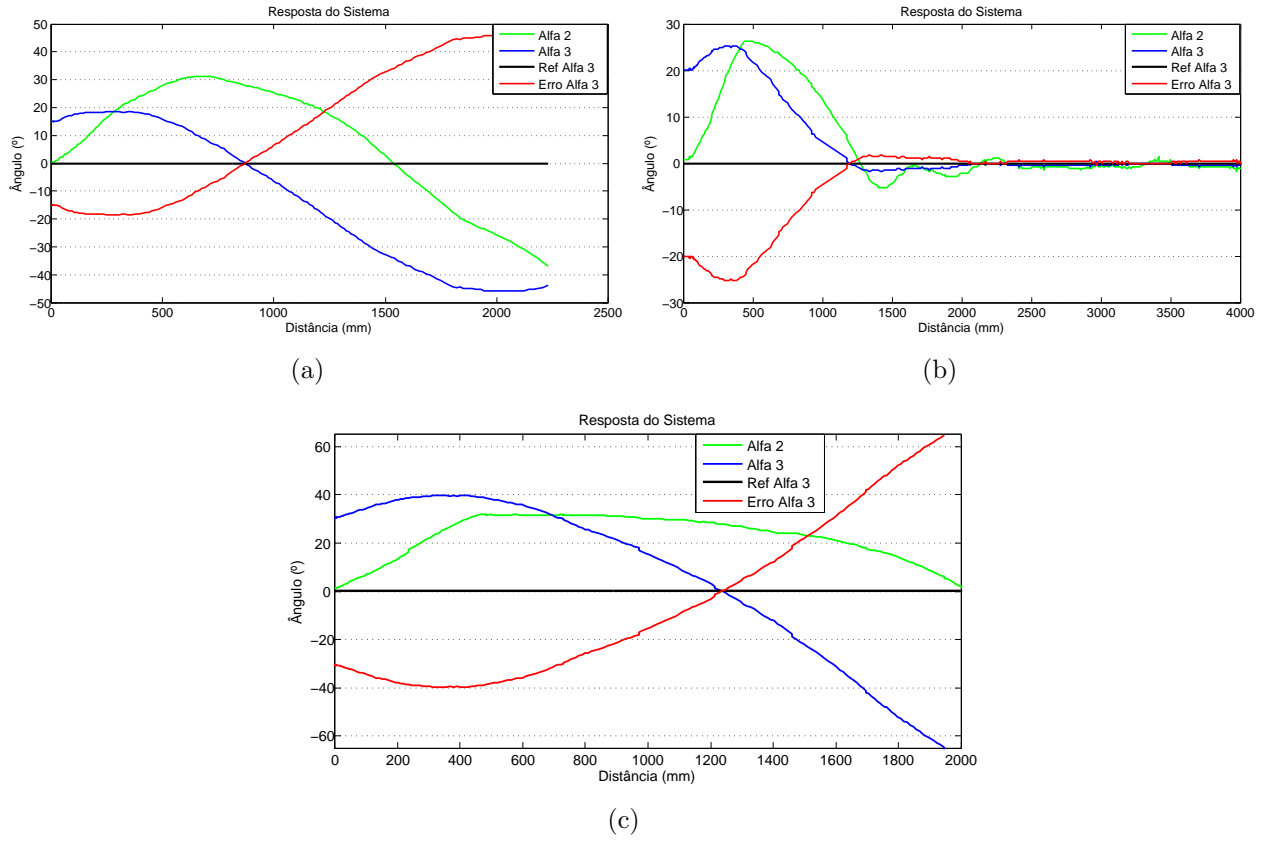


Figura 44: $Ref \alpha_3 = 0^\circ$ (a) Controlador Global - Engavetamento
(b) Controlado CGPR (c) Controlador CGPR - Engavetamento

o controlador Global. Observa-se que α_2 atingiu seu ângulo crítico, impossibilitando a continuidade da manobra. Contudo, a Figura 44(b) mostra que a referência foi atingida, mesmo tendo-se a situação inicial $[0^\circ, 20^\circ, \approx 0^\circ]$ que é mais severa que a testada com o controlador Global. Finalmente, a Figura 44(c) mostra que para a condição inicial $[0^\circ, 25^\circ, \approx 0^\circ]$ o controlador CGPR não conseguiu controlar a composição, pois, o ângulo α_3 atingiu seu valor crítico.

4.3.3 Experimentos III

O Experimento III consistiu em realizar ensaios com ambos controladores fornecendo-se diferentes referências para α_3 até que fosse alcançando o maior valor que ainda permite o controle do veículo. Nesse experimento a composição partia inicialmente alinhada e executava um movimento circular, caso a referência fosse atingida.

No primeiro ensaio desse experimento, foi fornecida uma referência de -20° . Como pode ser visto na Figura 45 ambos controladores atingiram o objetivo. Pode-se observar que o controlador CGPR atingiu a referência em uma distância menor, aproximadamente

700mm contra 1600mm do Global, além disso, obteve um erro em regime menor.

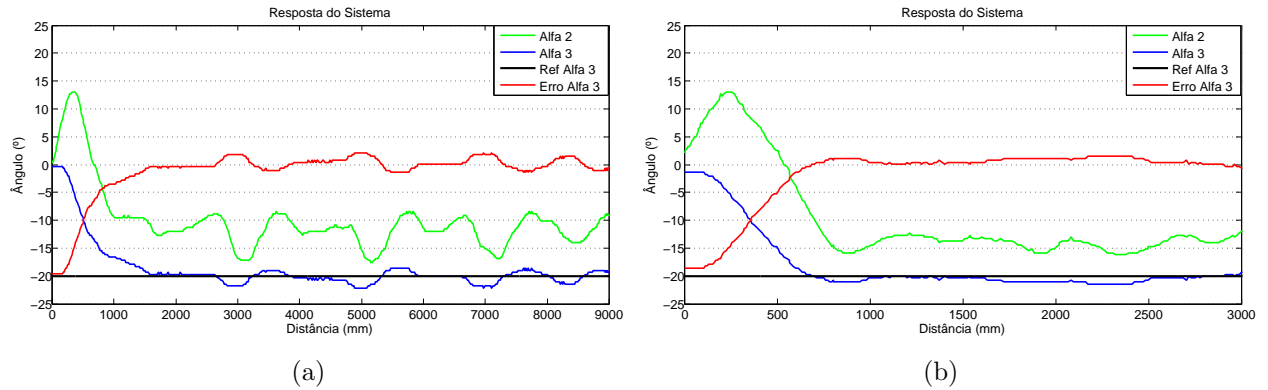


Figura 45: $Ref \alpha_3 = -20^\circ$ (a) Controlador Global (b) Controlado CGPR

No segundo teste foi fornecido uma referência de -30° . Conforme a Figura 46 mostra ambos controladores obtiveram velocidade de resposta semelhantes, porém, o controlador CGPR apresentou menor erro em regime, sendo aproximadamente zero.

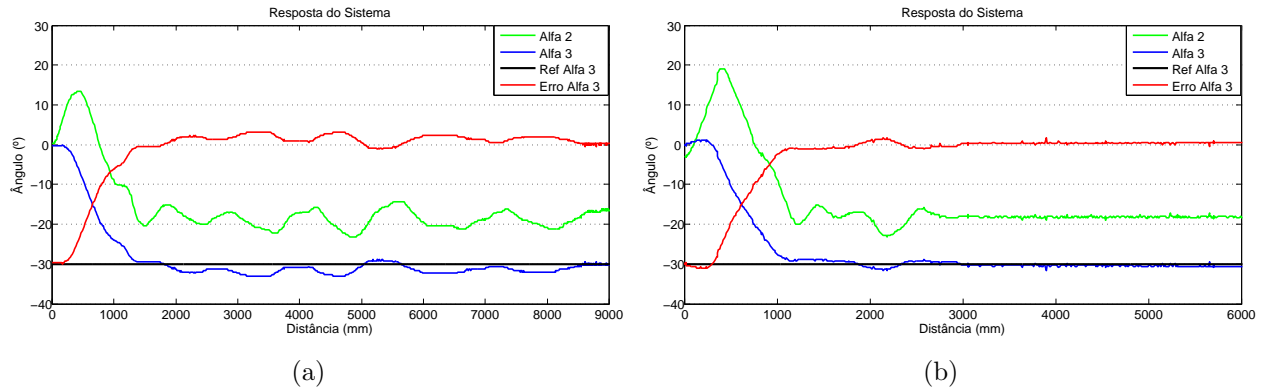


Figura 46: $Ref \alpha_3 = -30^\circ$ (a) Controlador Global (b) Controlado CGPR

A Figura 47 se refere ao terceiro ensaio, onde observa-se que para a referência de -35° o controlador Global, Figura 47(a), não controlou o veículo com o ângulo α_2 ultrapassando seu valor crítico. A Figura 47(b) mostra que o controlador CGPR permitiu que α_3 atingisse uma referência de -50° , porém não obtendo êxito para a referência de -60° , Figura 47(c).

No teste referente à Figura 47(c), pode-se ver claramente o efeito do atraso de configuração associado às taxas de variação dos ângulos das articulações. Quando o erro começou a crescer, em aproximadamente 1900mm, o controle inferiu que α_2 deveria diminuir em módulo, para isso devendo realizar o movimento de fechamento. Contudo, pelo fato de α_2 estar muito próximo ao seu valor crítico esse movimento é realizado muito

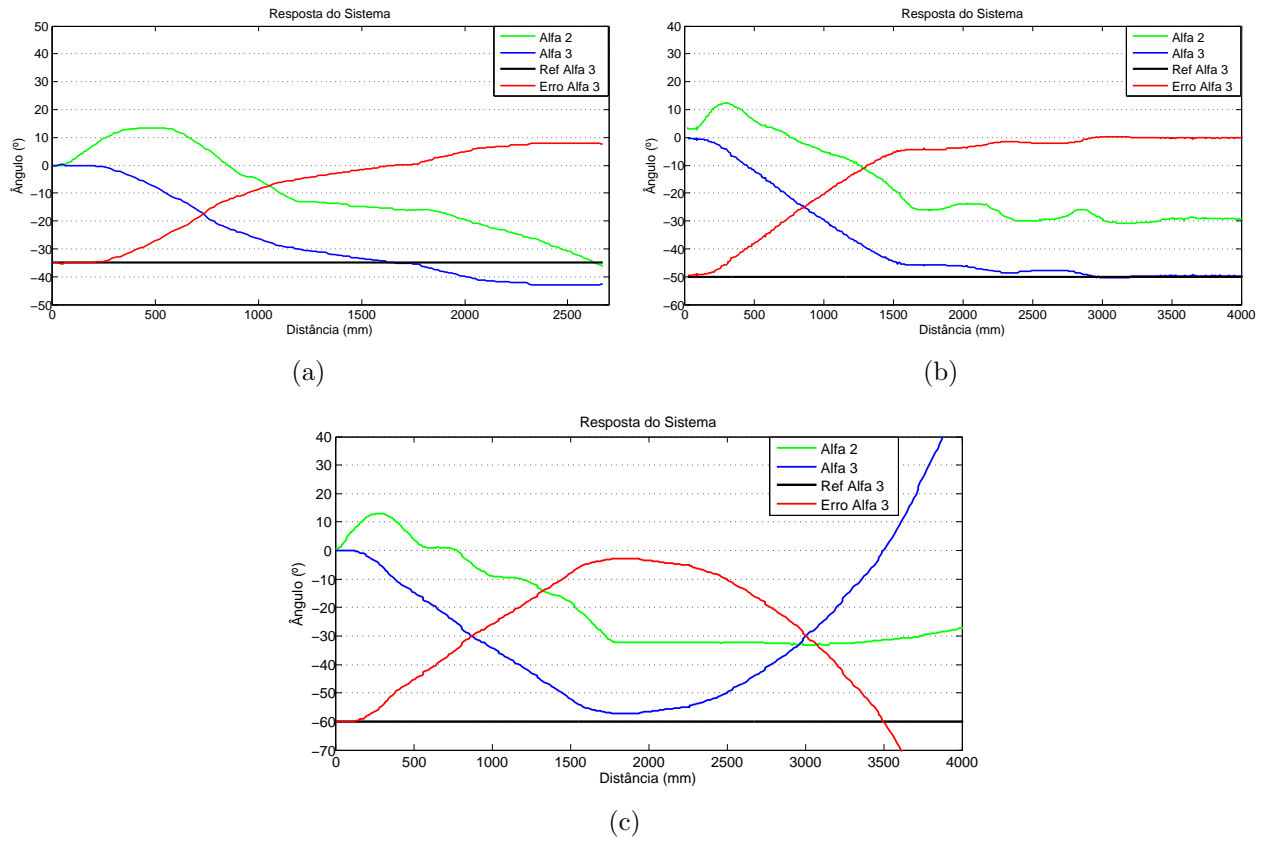


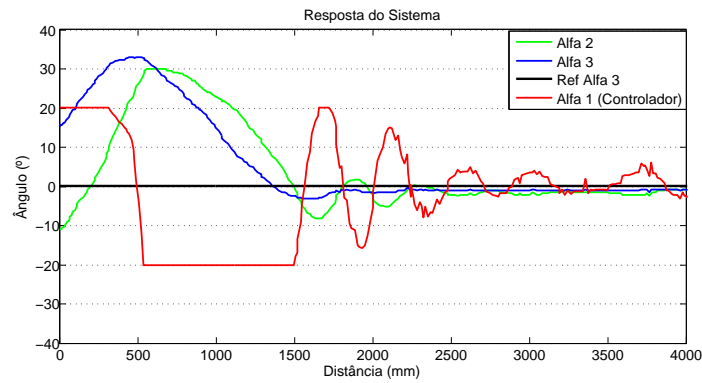
Figura 47: (a) Controlador Global - Engavetamento (b) Controlado CGPR
(c) Controlador CGPR - Engavetamento

lentamente não sendo suficiente para conter o crescimento de α_3 , levando esse último ao seu valor crítico.

4.3.4 Experimentos IV

Nesse experimento foi realizada uma manobra considerada crítica do ponto de vista de controle, essa manobra consiste em partir o veículo com os elementos da composição em formato de "Z", ou seja, com os ângulos das articulações possuindo sinais apostos e tendo-se como referência o alinhamento da composição, $Ref \alpha_3 = 0^\circ$.

A Figura 48, onde pode-se observar a ação de controle α_1 , mostra que para a condição $[0^\circ, \approx 18^\circ, \approx -10^\circ]$ o controlador CGPR conseguiu alinhar o veículo. Para valores de ângulos iniciais maiores que esses, o veículo entrava em *jackknife*. O controlador Global não foi capaz de realizar esse tipo de movimento.



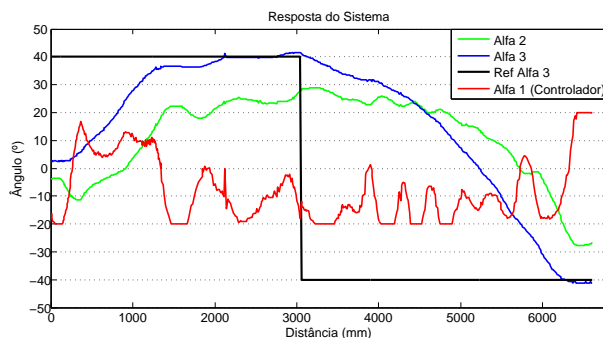
(a)

Figura 48: $Ref \alpha_3 = 0^\circ$ (a) Controlado CGPR

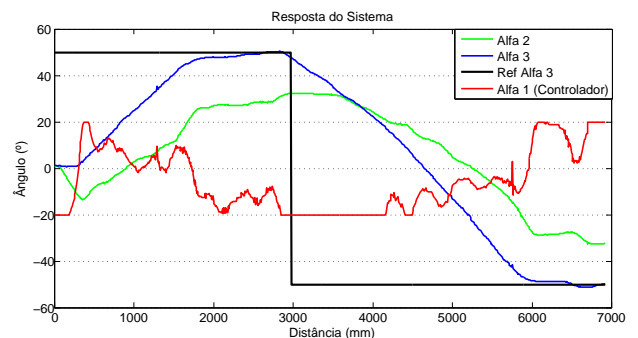
4.4 Conclusão

Mesmo sem o ajuste dos compensadores PID, utilizando $Kp = 1$, $Ki = 0$ e $Kd = 0$, o controlador CGPR apresentou resultados muito bons, quando comparado ao controlador Global. Após os ajustes dos compensadores, de modo geral, o controle se tornou mais preciso, rápido e estável. Devido à característica do atraso de configuração, existe um compromisso entre velocidade de resposta e faixa de manobrabilidade para o controle de configuração do veículo, que dever ser levado em consideração no desenvolvimento e ajustes de controladores de configuração.

Partindo-se de uma configuração não crítica e limitando-se as referências para α_3 em valores entre -50° e $+50^\circ$, o controlador CGPR permitiu a realização de navegação livre com o protótipo. A Figura 49 mostra um experimento de navegação em que são fornecidas duas referências consecutivas para α_3 . Observa-se que α_1 é saturado para valores maiores, em módulo, que 20° .



(a)



(b)

Figura 49: (a) $Ref \alpha_3 = 40^\circ$ seguido de $Ref \alpha_3 = -40^\circ$ (b) $Ref \alpha_3 = 50^\circ$ seguido de $Ref \alpha_3 = -50^\circ$

O controle de configuração não foi possível para referências de α_3 acima de 50° , isso se deve ao fato de que a medida que a referência se aproxima do ângulo crítico, a resposta da ação de controle se torna mais lenta e, portanto, não tolerando a presença de distúrbios, que são provenientes de diversas partes do processo. O aumento da precisão na quantificação dos ângulos e na atuação do mecanismo de direção, aliados a ajustes mais precisos dos ganhos dos compensadores, podem permitir que valores de referência maiores que 50° possam ser alcançados.

Além de avaliar os aspectos referentes ao controle de configuração de veículos articulados, os experimentos apresentados serviram também como teste para a própria plataforma desenvolvida, que se mostrou bastante apropriada para a realização dos experimentos, apesar das limitações de precisão do mecanismo de direção.

5 *Conclusões*

Nessa dissertação, foi desenvolvida uma plataforma experimental que possibilitou a realização de ensaios e validação de soluções para o problema de manobras e navegação de veículos ou robôs móveis multi-articulados. Além disso, foi apresentada uma estratégia *fuzzy* para o controle de configuração dos ângulos de V/RMA em movimento à ré. Essa solução de controle, denominado CGPR, foi avaliada e comparada a outras soluções existentes.

A plataforma experimental teve como base mecânica o automodelo em escala de um caminhão biarticulado. Para prover o veículo de capacidade de acionamento e instrumentação, foram desenvolvidos e confeccionados diferentes circuitos eletrônicos dotados de microcontroladores, sensores e atuadores eletro-mecânicos que foram devidamente embarcados. O *hardware* desenvolvido funcionou de forma satisfatória, cumprindo seu objetivo inicial.

Para permitir a troca de dados entre o computador remoto e o veículo foi desenvolvido um circuito de comunicação sem fio, que obteve desempenho adequado. O conceito adotado de processamento remoto, permite que diferentes técnicas de controle de manobra possam ser experimentadas sem a necessidade de qualquer modificação no hardware embarcado.

A etapa de desenvolvimento e preparação da plataforma experimental demandou a solução de problemas de diversos aspectos como a correção de folgas mecânicas, supressão de ruídos elétricos, concepção do protocolo de comunicação, elaboração de circuitos eletrônicos e programação, dentre outros. Assim, o conhecimento multidisciplinar foi uma característica presente nesse trabalho.

Outra relevante contribuição desse trabalho se deu no controle de configuração dos ângulos das articulações, através do CGPR. O controle de configuração é uma etapa fundamental para a obtenção da solução de navegação em geral de veículos e robôs móveis multi-articulados.

A partir dos resultados obtidos em [3], foi possível a obtenção da equação 3.4 que define o ângulo de giro para dois elementos consecutivos de um veículo multi-articulado. Tal equação é a base do conceito do controlador CGPR e permitiu uma abordagem de controle *fuzzy* auxiliado por técnicas analíticas.

O controlador CGPR, apresentou excelentes resultados, quando comparado a outras soluções anteriormente propostas. Além disso, devido a sua topologia em cascata, ele permite fácil expansão do número de elementos passivos da composição, mantendo um custo computacional intermediário entre os controladores CpPI e Global.

Ainda como contribuição na área de controle, foram abordados aspectos relacionados à dinâmica do movimento à ré de V/RMA, sobretudo em relação ao atraso de configuração, que foi caracterizado e teve soluções propostas para minimizar seus efeitos.

Por fim, além das contribuições citadas, esse trabalho possibilitou, além dessa dissertação de mestrado, a realização de diversos trabalhos acadêmicos, sendo eles: uma tese de mestrado [13], com outras duas em andamento, três projetos de graduação [1, 12, 14] e cinco artigos publicados [46, 47, 50–52].

5.1 Trabalhos Futuros

A realização desse trabalho abre espaço para que diversas melhorias na plataforma experimental e na solução de controle de configuração possam ser realizadas através de trabalhos futuros. A seguir serão sugeridos alguns pontos que podem ser melhorados e investigados.

- Aumento do ângulo máximo das rodas de direção limitado pelo protótipo. Essa modificação melhora a manobrabilidade e expande os limites de controle do veículo, sendo importante sobretudo caso seja adicionado um novo elemento na composição.
- Reduzir o passo de discretização de atuação nas rodas de direção para um valor menor que o atual de 1° , o que forneceria maior precisão ao sinal de controle.
- Melhoria do sistema de odometria através da instalação, em local mais apropriado, de um *encoder* com maior relação pulsos/volta, permitindo-se assim, a melhoria do sinal de realimentação para o controle de velocidade do veículo.
- Otimizar o algoritmo de comunicação com o *joystick*, o que permitirá o uso desse dispositivo na geração de referências angulares para α_3 de modo eficiente.

- Implementar o código de controle de configuração em uma linguagem com melhor desempenho que o MATLAB[®], como por exemplo *C*. Essa medida, associada a uma comunicação sem fio mais veloz, aumentará a taxa de atualização do controle permitindo que o veículos possa se deslocar em maiores velocidades.
- Determinação dos ganhos ótimos dos compensadores PID através da identificação do erro por meio da excitação da planta com um sinal PRBS (PseudoRandom Binary Sequence). Esse ajuste fino dos ganhos de cada controlador, pode ser realizado com a presença simultânea dos dois *trailers* na composição, o que incorpora a influência dos elementos entre si.
- Melhorar os resultados obtidos pelo controlador CGPR, por meio de ajustes dos parâmetros do sistema *fuzzy* e dos compensadores.
- Desenvolver soluções para a etapa responsável pelo seguimento de trajetórias. Essa etapa tem como principal função a geração de referências para a etapa de controle de configuração.
- Desenvolver algoritmos para a determinação de trajetórias ótimas para veículos multi-articulados. Essa etapa permitirá identificar prematuramente, por meio de simulação baseada no modelo analítico do veículo, se existe a possibilidade do destino ser alcançado, caso exista, a etapa de seguimento de trajetória se encarregará de executá-la.
- Desenvolver sistemas para a detecção de vagas para estacionamento e obstáculos. Essa é a etapa hierarquicamente superior do problema de navegação de veículos ou robôs móveis multi-articulados. A identificação das restrições espaciais ao redor do veículo, que pode ser realizado por meio de câmeras ou sensores embarcados, permitirá a definição da trajetória que deve ser seguida.

Referências

- [1] PANDOLFI, F. *Proposta de Controle de Robô Móvel Multi-Articulado Usando um Controlador Fuzzy Global Gerado a Partir de Exemplos*. 2009. Monografia para graduação em Engenharia Computação, Universidade Federal do Espírito Santo.
- [2] WIDROW, B. et al. Adaptive inverse control based on nonlinear adaptive filtering. In: *Proceedings of 5th IFAC Workshop on Algorithms and Architectures for Real-Time Control (AARTC'98)*. Cancun, Mexico: Elsevier Science Ltd, Oxford, UK, 1998.
- [3] KULITZ, H. R. *Modelagem e Controle Fuzzy de Robôs e Veículos Multi-Articulados*. Tese (Doutorado) — Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Espírito Santo, Vitória, 2003.
- [4] AGÊNCIA T1. Disponível em: <<http://www.agenciat1.com.br/1710-bb-simplifica-compra-de-caminhoes-novos-e-usados/>>. Acesso em: 02 ago 2010.
- [5] PINHEIRO, M. P. L. *Simulação do Movimento de Robôs Móveis Multi-Articulados e Obtenção de um Preditor Fuzzy*. 130f. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia Mecânica, Universidade Federal do Espírito Santo, Vitória, 2004.
- [6] XU, J.; CHEN, G.; XIE, M. Vision-guided automatic parking for smart car. *IEEE Intelligent Vehicles Symposium*, IV, p. 725–730, October 2000.
- [7] LI, T.-H. S.; CHANG, S.-J. Autonomous fuzzy parking control of car-like mobile robot. *IEEE Transactions on System, Man and Cybernetics - Part A: Systems and Humans*, v. 33, n. 4, p. 451–465, Julho 2003.
- [8] BROWN, N. H.; WOOD, H. C.; WILSON, J. N. Image analysis for vision-based agricultural vehicle guidance. In: SPIE. *Proceedings of SPIE Optics in Agriculture*. Boston, MA, USA: SPIE, 1991. (Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, v. 1379), p. 54–68.
- [9] NGUYEN, T. M.; SINKO, J. W.; GALIJAN, R. C. Using differential carrier phase gps to control automated vehicles. *IEEE Transactions on System, Man and Cybernetics - Part A: Systems and Humans*, Sacramento, CA, USA, v. 1, n. 4, p. 493–496, Agosto 1997.
- [10] JUNG, C. R. et al. Computação embarcada: Projeto e implementação de veículos autônomos inteligentes. In: *Anais do XXIV Congresso da Sociedade Brasileira de Computação. Mini-Curso: Jornada de Atualização em Informática*. São Leopoldo, RS: SBC, 2005. v. 1, p. 1358–1406.

- [11] DAXWANGER, W. A.; SCHMIDT, G. K. Skill-based visual parking control using neural and fuzzy networks. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*. Vancouver, BC, Canada: IEEE, 1995. v. 2, p. 1659–1664.
- [12] MELLO, D. N. de. *Estacionamento Automático de Veículos Usando Lógica Fuzzy*. 2008. Monografia para graduação em Engenharia da Computação, Universidade Federal do Espírito Santo.
- [13] RAMPINELLI, M. F. *Navegação de Robôs Móveis ou Veículos Multi-Articulados com Realimentação de Informações Visuais*. 84f. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, 2008.
- [14] MARQUES, V. M. *Técnica de Modelagem e Controle Neural Inverso do Movimento à Ré de um Veículo Articulado Complexo*. 2009. Monografia para graduação em Engenharia da Computação, Universidade Federal do Espírito Santo.
- [15] SILVA, E. B. *Modelagem e Controle Neural Inverso dos Movimentos de um Veículo Articulado Complexo*. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, 2003.
- [16] FERREIRA, E. P.; LAMEGO, M. M.; WIDROW, B. Neurointerfaces for semi-autonomous object moving systems. In: *Proceedings of the 14th World IFAC Congress*. Bejin, China: Elsevier Science Ltd, Oxford, UK, 1999. K, p. 155–160.
- [17] PARRA-LOERA, R.; CORELIS, D. J. Expert system controller for backing-up a truck-trailer system in a constrained space. In: *Proceedings of the 37th Midwest Symposium on Circuits and Systems*. Lafayette, LA, USA: IEEE Press, 1994. v. 2, p. 1357–1361.
- [18] YI, J.; YUBAZAKI, N.; HIROTA, K. Backing up control of truck-trailer system. In: *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*. Melbourne, Australia: IEEE Press, 2001. v. 1, p. 489–492.
- [19] HALGAMUGE, S. K.; RUNKLER, T. A.; GLESNER, M. A hierarchical hybrid fuzzy controller for real-time reverse driving support of vehicles with long trailers. In: *Proceedings of the Third IEEE Conference on Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence*. Orlando, FL, USA: IEEE, 1994. v. 2, p. 1207–1210.
- [20] KONG, S.-G.; KOSKO, B. Adaptive fuzzy systems for backing up a truck-and-trailer. *IEEE Transactions on Neural Networks*, v. 3, n. 2, p. 211–223, Mar 1992.
- [21] SAMPEI, M. et al. Arbitrary path tracking control of articulated vehicles using non-linear control theory. *IEEE Transactions on Control Systems Technology*, IEEE, v. 3, n. 1, p. 125–131, Mar 1995.
- [22] LAMIRAUX, F.; LAUMOND, J. P. A practical approach to feedback control for a mobile robot with trailer. In: *Proceedings of IEEE International Conference on Robotics and Automation*. Leuven, Belgium: IEEE, 1998. v. 4, p. 3291–3296.

- [23] NAKAMURA, Y. et al. Design of steering mechanism and control of nonholonomic trailer systems. In: *Proceedings of IEEE International Conference on Robotics and Automation*. San Francisco, CA, USA: IEEE, 2000. p. 247–254.
- [24] KIM, D.-H.; OH, J.-H. Experiments of backward tracking control for trailer system. In: *Proceedings of IEEE International Conference on Robotics and Automation*. Detroit, MI: IEEE, 1999. v. 1, p. 19–22.
- [25] WIDMER, J. A. Compatibilidade de tráfego de bitrens de 25m com a infra-estrutura viária brasileira. *2º Colloquium Internacional de Suspensões*, 2002.
- [26] TAMIYA Home Page. Disponível em: <<http://www.tamiya.com>>. Acesso em: 11 Mar 2010.
- [27] CORPORATION, A. *ATMega32 Datasheet*. [S.l.]. Disponível em: <<http://www.atmel.com>>.
- [28] CADSOFT Home Page. Disponível em: <<http://www.cadsoftusa.com/freeware.htm>>. Acesso em: 27 Abr 2010.
- [29] J., E. H. R. B.; FENG, L. *Systems and Methods for Mobile Robot Positioning*. 1nd. ed. University of Michigan, USA: J. Borenstein, 1996.
- [30] FUTABA Radio Control. Disponível em: <<http://www.futaba-rc.com>>. Acesso em: 12 Jan 2010.
- [31] AUTOMÁTICA Tecnologia Home Page. Disponível em: <<http://www.automatica.com.br/>>. Acesso em: 4 Mai 2010.
- [32] SHARP Microelectronics of the Americas. Ago 2009. Disponível em: <<http://www.sharpsma.com/Page.aspx/americas/en/part/GP2D120/>>. Acesso em: 20 Ago 2009.
- [33] M., R. D.; W., K. B. *C A Linguagem De Programacao Padrao Ansi*. 1nd. ed. [S.l.]: CAMPUS, 1989.
- [34] HPINFOTECH Home Page. Disponível em: <<http://www.hpinfotech.ro/>>. Acesso em: 15 Jun 2010.
- [35] RF Solutions. Disponível em: <<http://www.rfsolutions.co.uk/acatalog/index.html>>. Acesso em: 04 Ago 2010.
- [36] LAIPAC Tec. Disponível em: <<http://www.laipac.com>>. Acesso em: 05 Ago 2010.
- [37] KUO, B. C. *Automatic control systems*. 8a. ed.. ed. New York: John Wiley and Sons, 2003.
- [38] CHEN, C.-T. *Linear System Theory and Design*. 3a. ed.. ed. New York: Oxford, 1999.
- [39] ROJAS, R. *Neural Networks: A System Introduction*. 1. ed. Berlin, New York: Springer-Verlag, 1996.

- [40] ZADEH, L. A. Outline of a new approach to the analysis of complex system and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 3, n. 1, p. 28–44, 1973.
- [41] ZADEH, L. A. The concept of a linguistic variable and its applications to approximate reasoning - part i. *Information Sciences*, v. 8, n. 1, p. 199–249, 1975.
- [42] ZADEH, L. A. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, v. 1, p. 3–28, 1978.
- [43] SHAW, I. S.; SIMÕES, M. G. *Modelagem e Controle Fuzzy*. 1a. ed.. ed. São Paulo, SP: Edgard Blücher, 1999.
- [44] ALMEIDA, P. E. M. de; EVSUKOFF, A. G. Sistemas fuzzy. In: SOLANGE OLIVEIRA REZENDE (ORG.). *Sistemas Inteligentes: Fundamentos e Aplicações*. 1. ed. SP: Editora Manole, 2003. p. 169–202.
- [45] WANG, L.-X.; MENDEL, J. M. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 22, n. 6, p. 1414–1427, 1992.
- [46] FERREIRA, F. P. E. P.; REINAN, T. Modelagem fuzzy de robôs móveis multi-articulados em manobras complexas: Descrição de movimentos e controladores reais, em malha aberta. *SBAI*, 2009.
- [47] REINAN, E. P. F. T.; PANDOLFI, F. Controle fuzzy global e desacoplado, para robôs e/ou veículos multi-articulados: Caso real truck-trailer-trailer. *SBAI*, 2009.
- [48] MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, Academic Press, Inc., Duluth, MN, USA, v. 7, n. 1, p. 135–147, Jan 75.
- [49] MATHWORKS Home Page. Disponível em: <<http://www.mathworks.com/>>. Acesso em: 15 Jul 2010.
- [50] FERREIRA, F. P. E. P.; REINAN, T. Modelagem fuzzy de robôs móveis multi-articulados em manobras complexas com atraso: Uma nova sistemática para a descrição de movimentos e controladores. *CBA*, 2010.
- [51] RAMPINELLI, E. P. F. M.; REINAN, T. Análise da robutez da visão computacional de topo no controle de movimento de robôs móveis ou veículos multi-articulados. *Induscon*, 2008.
- [52] REINAN, M. R. T.; FERREIRA, E. P. Análise comparativa do uso de visão artificial versus sensores embarcados no controle de manobras em robôs móveis multi-articulados. *CBA*, 2008.

ANEXO A – Estacionamento de veículo não articulados

1 PROGRAMA DE CONTROLE

Para realizar as manobras de estacionamento do carro, é necessária a construção de um programa que realize as seguintes funções:

- Definir orientação inicial – o carro identifica a orientação inicial em relação à horizontal, podendo ainda realizar manobras de ajuste de forma que o carro pare paralelo à horizontal.
- Procurar vaga de estacionamento – o carro se movimenta para frente passando pela vaga e detectando as bordas da vaga ou dos espaços vazios.
- Definir o ambiente – determina as coordenadas usadas para descrever o carro e a vaga (ponto final de estacionamento).
- Definir trajetória – determina a curva de seguimento desejada para realizar o estacionamento ou as aproximações da curva.
- Estacionar carro – executa as manobras para o estacionamento à ré utilizando uma das estratégias definidas, que são: aproximação por distância e aproximação por tempo.

1.1 DEFINIR ORIENTAÇÃO INICIAL

Esta ação do controlador faz uma leitura dos sensores do carro e acha a distância e a orientação do carro em relação às laterais. Os valores efetivamente medidos podem ser vistos na figura abaixo.

A distância m é a distância entre os sensores, d_1 e d_2 são as medições realizadas pelos sensores s_1 e s_2 . Pela geometria, o ângulo theta é calculado da seguinte forma:

$$\theta = \arctan\left(\frac{d_1 - d_2}{m}\right)$$

Onde

$$\theta = \begin{cases} \geq 0, & d_1 \geq d_2 \\ < 0, & d_1 < d_2 \end{cases}$$

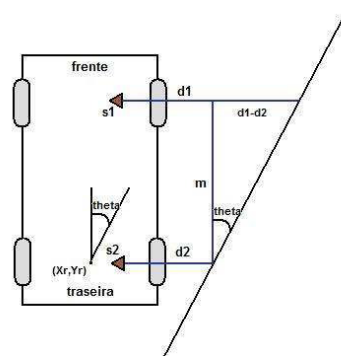


Figura 42 – Distâncias e medições de ângulo

Foram feitas quatro coletas para exemplificar os resultados. Na primeira medição, as distâncias e o resultado são:

Tabela 3 – Resultados para a primeira medição real

Sensores	Posição Real [mm]	Ângulo Real [graus]	Medição [mm]	Ângulo Medição [graus]
Sensor S1	193,00	0,19	197,06	-1,04
Sensor S2	192,00		202,61	

Na segunda medição, as distâncias e os resultados são:

Tabela 4 – Resultados para a segunda medição real

Sensores	Posição Real [mm]	Ângulo Real [graus]	Medição [mm]	Ângulo Medição [graus]
Sensor S1	270,00	16,96	291,67	19,62
Sensor S2	177,00		182,94	

Na terceira medição, as distâncias e os resultados são:

Tabela 5 - Resultados para a terceira medição real

Sensores	Posição Real [mm]	Ângulo Real [graus]	Medição [mm]	Ângulo Medição [graus]
Sensor S1	173,00	-38,31	175,00	-36,42
Sensor S2	414,00		400,00	

Na quarta medição, as distâncias e os resultados são:

Tabela 6 - Resultados para a quarta medição real

Sensores	Posição Real [mm]	Ângulo Real [graus]	Medição [mm]	Ângulo Medição [graus]
Sensor S1	362,00	-11,13	365,71	-7,44
Sensor S2	422,00		405,56	

Abaixo as medições feitas em sentido horário.

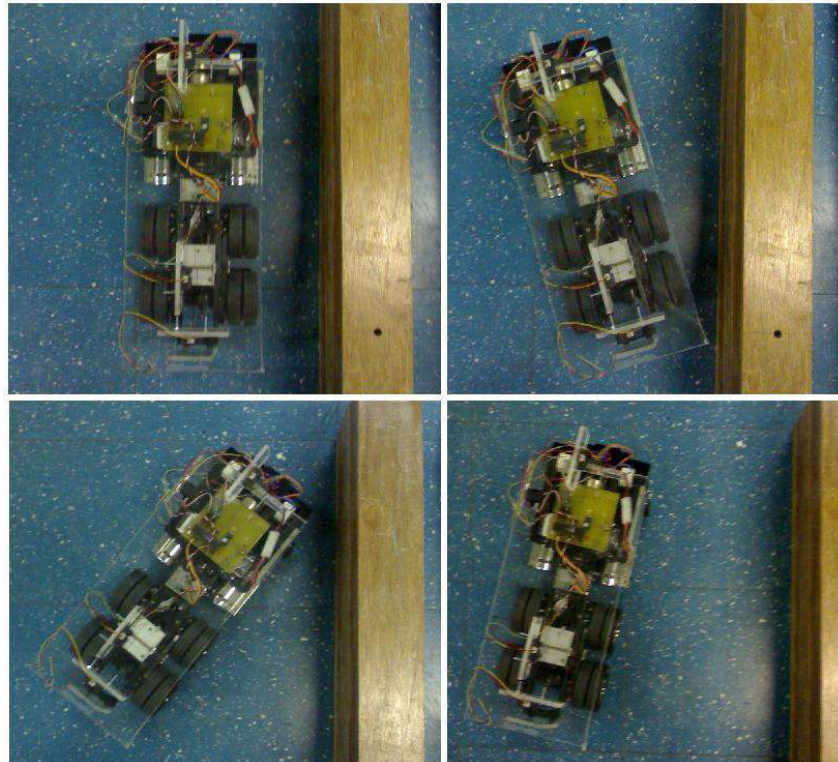


Figura 43 – Definir a orientação inicial para quatro configurações propostas. Em sentido horário, da primeira à quarta medição.

1.1.1 AJUSTAR POSIÇÃO INICIAL

Como alternativa, pode ser construída uma função que reorienta o carro para que este fique próximo a parede lateral e com ângulo igual a zero.

Em [1] é descrito um controlador fuzzy para aproximação a paredes e ajustes finais dentro da vaga de estacionamento. Tal função não foi implementada neste trabalho e pode ser inserida em trabalhos futuros.

1.2 PROCURAR VAGA DE ESTACIONAMENTO

Esta ação movimenta o carro para que a vaga seja identificada. Os dois sensores laterais devem achar as bordas da vaga de estacionamento e a partir das informações das bordas, calcularem o tamanho da vaga e sua profundidade.

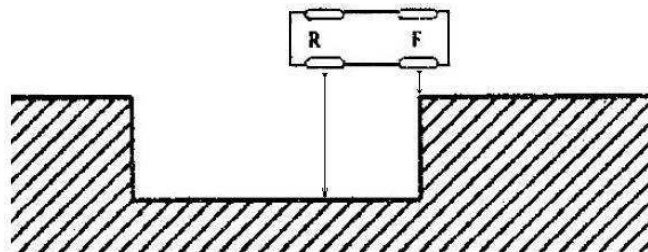


Figura 44 - Detecção das Bordas da Vaga de Estacionamento

Considerando os sensores laterais s_1 e s_2 , foram declaradas duas variáveis chamadas $flagS1$ e $flagS2$ que representam a detecção de uma borda de vaga. As variáveis $flagS1$ e $flagS2$ reconhecem a borda (valor definido como *UP*) quando a distância medida for maior ou igual à largura do carro vezes uma constante. Caso a distância fique menor do que essa multiplicação, os sensores não reconhecem uma borda (valor definido como *DOWN*).

Os estados usados para o reconhecimento da vaga podem ser descritos conforme figura abaixo. Os estados inicial e final são S_0 e S_4 . A transição entre os estados intermediários faz com que a vaga tenha largura e comprimento adequados ao estacionamento. O parâmetro usado para definir o comprimento da vaga segue exemplo da largura, é igual ao comprimento do carro vezes uma constante.

A transição dos estados conforme mostrada na figura faz com que os intervalos entre carros não sejam reconhecidos como vagas.

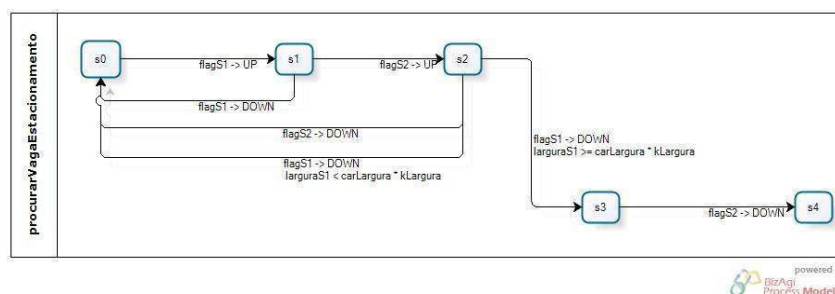


Figura 45 – Transições dos estados para reconhecimento da vaga de estacionamento

Foram feitas 4 coletas da função “Procurar Vaga de Estacionamento”, testando tamanhos de vagas menores e maiores do que uma vaga estacionável. As limitações impostas pelo carro influenciam os resultados de algumas dessas medições, conforme será informado.

Na primeira medição, a velocidade linear definida foi 218,5 mm/s e o comprimento da vaga foi de 400 mm. O comprimento medido da vaga foi de 340,4 mm. Como a velocidade linear do carro não é constante e depende da quantidade de carga na bateria, a velocidade precisa ser corrigida.

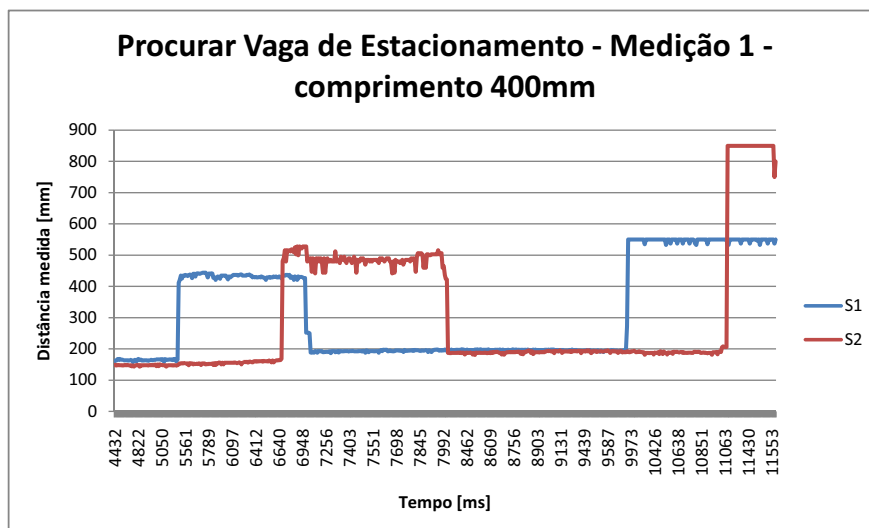


Figura 46 – Procurar Vaga – medição 1 – velocidade = 218,5 mm/s e comprimento = 400 mm

O reconhecimento das bordas acontece na primeira elevação de s_1 e s_2 . A largura da vaga é de 445 mm. Como o comprimento não era suficiente para fazer o carro parar de se movimentar, pois nenhuma vaga estacionável havia sido reconhecida, foi colocado além do limite da vaga um outro objeto para que a distância necessária fosse reconhecida.

Na segunda medição, a velocidade linear definida foi 218,5 mm/s e o comprimento da vaga foi de 500 mm. O comprimento medido da vaga foi de 417,1 mm. Com os dois primeiros resultados, a velocidade foi corrigida para 262,5 mm/s.

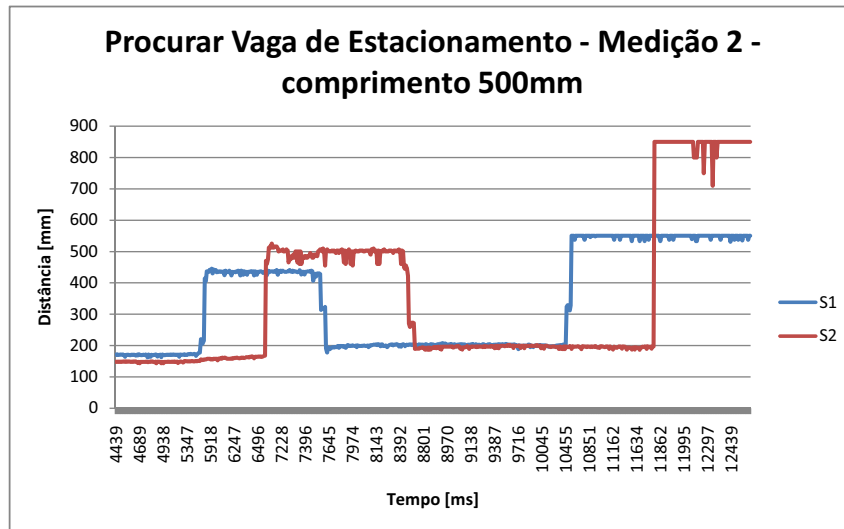


Figura 47 – Procurar Vaga – medição 2 – velocidade = 218,5 mm/s e comprimento = 500 mm

As distâncias medidas após o não reconhecimento da vaga, trecho entre os instantes 7645 ms e 10455 ms para s_1 e entre os instantes 8801 ms e 11865 ms, aparecem muito elevadas e diferentes devido à limitação de distância reconhecida dos sensores. Neste caso, o carro passa por um trecho onde os sensores não percebem objetos próximos.

Na terceira medição, a velocidade linear definida foi 262,5 mm/s e o comprimento da vaga foi de 700 mm. O comprimento medido da vaga foi de 691,7 mm.

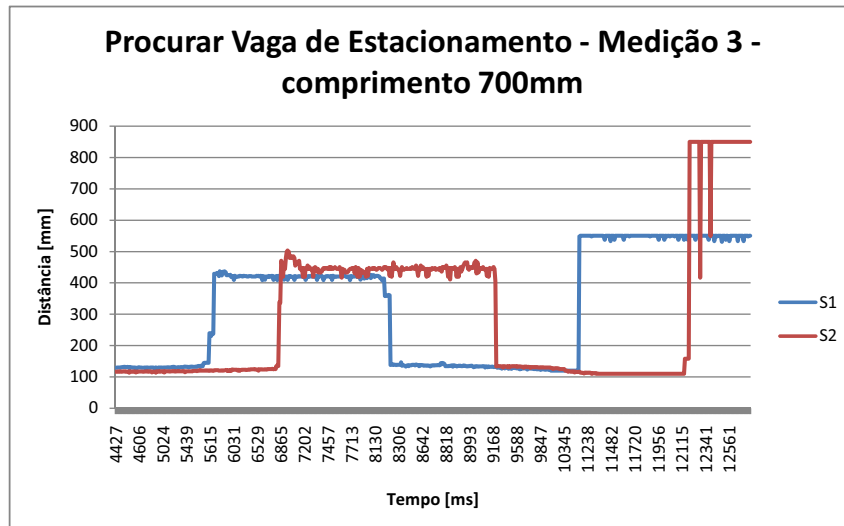


Figura 48 - Procurar Vaga – medição 3 – velocidade = 262,5 mm/s e comprimento = 700 mm



Figura 49 – Vaga de estacionamento e obstáculo no final para condição de parada do carro

Na última medição, a velocidade linear definida foi 262,5 mm/s e o comprimento da vaga foi de 800 mm. O comprimento medido da vaga foi de 799,0 mm.

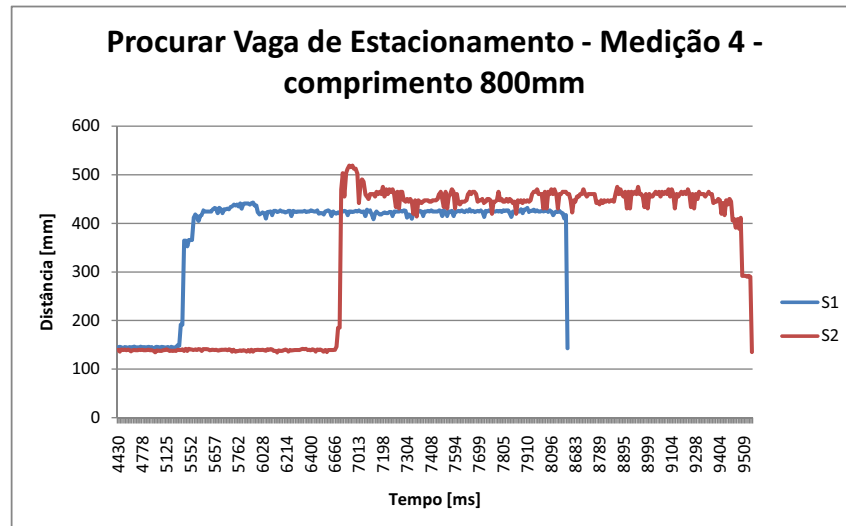


Figura 50 – Procurar Vaga – medição 4 – velocidade = 262,5 mm/s e comprimento = 800 mm



Figura 51 – Vaga reconhecida, distância de 800 mm medida como 799,0 mm

1.3 DEFINIR AMBIENTE

Esta ação, a partir da ação anterior, define o ambiente do estacionamento, criando as coordenadas X e Y utilizadas para o movimento do carro.

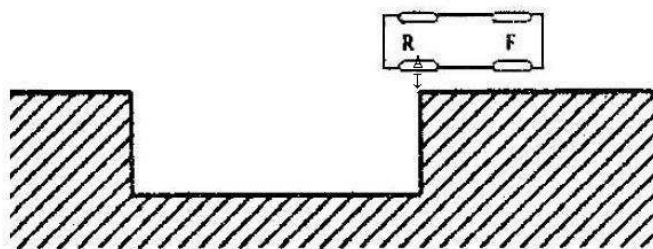


Figura 52 – Carro posicionado na fronteira da vaga para estacionamento

Ao chegar ao estado s_4 da ação anterior, o carro recebe o comando para parar. No entanto, há um atraso na medição dos sensores e 38 ms, ou seja, a borda reconhecida pode ter passado a 38 ms, que com a velocidade utilizada na última medição representa 9,98 mm.

O ponto final do estacionamento está localizado, em relação a ponto atual do carro, a distância de “largura do carro” + “distância até a parede próxima medida” no eixo Y e a distância de “distância da frente do carro até o eixo dianteiro” + “distância entre eixos” + “fator sobre o comprimento do carro”.



Figura 53 – Posição final do carro para cálculo de (X_e, Y_e) , início do estacionamento

O ponto de interesse nesse cálculo é o (x_e, y_e) , uma vez que com ele traçamos a curva de estacionamento.

1.4 DEFINIR TRAJETÓRIA

Segundo [1] e [3], a curva que oferece maiores vantagens ao estacionamento paralelo à mão de direção é o polinômio de quinta ordem representado abaixo:

$$y_r(x_r) = a_5 \left(\frac{x_r}{x_e} \right)^5 + a_4 \left(\frac{x_r}{x_e} \right)^4 + a_3 \left(\frac{x_r}{x_e} \right)^3 + a_2 \left(\frac{x_r}{x_e} \right)^2 + a_1 \left(\frac{x_r}{x_e} \right) + a_0$$

Equação 2 – Equação geral para o estacionamento paralelo à mão de direção

O polinômio é construído de tal forma que a posição inicial seja o ponto (x_e, y_e) e o ponto final seja $(0,0)$. Assim, $a_0 = 0$. As restrições para a determinação dos coeficientes da equação são:

$y'_r(x_e) = y'_r(0) = 0$, ou seja, o carro no momento inicial e final faz movimentos paralelos ao eixo X;

$k(x_e) = k(0) = 0$, ou seja, o carro não está realizando curvas nos momentos inicial e final da trajetória. $k(x_r)$ é a curvatura da trajetória seguida pelo modelo e é definida como:

$$k(x_r) = \frac{y''_r}{[1 + (y'_r)^2]^{\frac{3}{2}}}$$

Equação 3 – Curvatura da trajetória

Calculando a primeira e a segunda derivada da equação 1, temos:

$$y'_r(x_r) = \frac{5a_5}{x_e} \left(\frac{x_r}{x_e} \right)^4 + \frac{4a_4}{x_e} \left(\frac{x_r}{x_e} \right)^3 + \frac{3a_3}{x_e} \left(\frac{x_r}{x_e} \right)^2 + \frac{2a_2}{x_e} \left(\frac{x_r}{x_e} \right) + \frac{a_1}{x_e}$$

Equação 4 – Primeira derivada da equação geral

$$y''_r(x_r) = \frac{20a_5}{x_e^2} \left(\frac{x_r}{x_e} \right)^3 + \frac{12a_4}{x_e^2} \left(\frac{x_r}{x_e} \right)^2 + \frac{6a_3}{x_e^2} \left(\frac{x_r}{x_e} \right) + \frac{2a_2}{x_e^2}$$

Equação 5 – Segunda derivada da equação geral

Considerando a restrição 1 $y'_r(0) = 0$, temos:

$$y'_r(0) = 0 = \frac{5a_5}{x_e} (0)^4 + \frac{4a_4}{x_e} (0)^3 + \frac{3a_3}{x_e} (0)^2 + \frac{2a_2}{x_e} (0) + \frac{a_1}{x_e}$$

$$a_1 = 0$$

Resultado 1 - Coeficiente da equação geral

Considerando a restrição 2, $k(0) = 0$, temos:

$$k(0) = 0 = \frac{y''_r}{[1 + (y'_r)^2]^{\frac{3}{2}}}$$

$$y''_r(0) = 0 = \frac{20a_5}{x_e^2} (0)^3 + \frac{12a_4}{x_e^2} (0)^2 + \frac{6a_3}{x_e^2} (0) + \frac{2a_2}{x_e^2}$$

$$a_2 = 0$$

Resultado 2 - Coeficiente da equação geral

Considerando as restrições 2 e 1, $k(x_e) = 0$ e $y'_r(x_e) = 0$, temos:

$$k(x_e) = 0 = \frac{y_r''}{[1 + (0)^2]^{\frac{3}{2}}}$$

$$y_r''(x_e) = 0 = \frac{20a_5}{x_e^2}(1)^3 + \frac{12a_4}{x_e^2}(1)^2 + \frac{6a_3}{x_e^2}(1)$$

$$10a_5 + 6a_4 + 3a_3 = 0$$

Resultado 3 – Coeficientes da equação geral

Substituindo os resultados 1, 2 e 3 na equação 2 para o ponto (x_e, y_e) , temos:

$$y_r'(x_e) = 0 = \frac{5a_5}{x_e}(1)^4 + \frac{4a_4}{x_e}(1)^3 + \frac{3a_3}{x_e}(1)^2 + 0(1) + 0$$

$$5a_5 + 4a_4 + 3a_3 = 0$$

$$5a_5 = -(4a_4 + 3a_3)$$

Resultado 4 - Coeficientes da equação geral

Substituindo 4 em 3, temos:

$$-2(4a_4 + 3a_3) + 6a_4 + 3a_0 = 0$$

$$2a_4 = -3a_3$$

Resultado 5 – Coeficientes da equação geral

Substituindo os resultados 1, 2, 3, 4 e 5 na equação 1 para o ponto (x_e, y_e) , temos:

$$y_r(x_e) = a_5(1)^5 + a_4(1)^4 + a_3(1)^3 + 0(1)^2 + 0(0) + 0$$

$$y_e = \frac{3a_3}{5} - \frac{3a_3}{2} + a_3 = \frac{(6a_3 - 15a_3 + 10a_3)}{10}$$

$$a_3 = 10y_e$$

Resultado 6 – Coeficiente da equação geral

Substituindo o resultado 6 nos resultados 5 e 4, temos:

$$a_4 = -15y_e \text{ e } a_5 = 6y_e$$

Resultado 7 – Coeficientes da equação geral

Substituindo os resultados 6 e 7 nas equações 1, 3 e 4, temos:

$$y_r(x_r) = y_e \left[6 \left(\frac{x_r}{x_e} \right)^5 - 15 \left(\frac{x_r}{x_e} \right)^4 + 10 \left(\frac{x_r}{x_e} \right)^3 \right]$$

Equação 6 – Equação da trajetória do modelo, dado um ponto inicial (x_e, y_e)

$$y_r'(x_r) = \frac{30y_e}{x_e} \left[\left(\frac{x_r}{x_e} \right)^4 - 2 \left(\frac{x_r}{x_e} \right)^3 + \left(\frac{x_r}{x_e} \right)^2 \right]$$

Equação 7 – Primeira derivada da trajetória do modelo, dado um ponto inicial (x_e, y_e)

$$y_r''(x_r) = \frac{60y_e}{x_e^2} \left[2 \left(\frac{x_r}{x_e} \right)^3 - 3 \left(\frac{x_r}{x_e} \right)^2 + \left(\frac{x_r}{x_e} \right) \right]$$

Equação 8 - Segunda derivada da trajetória do modelo, dado um ponto inicial (x_e, y_e)

$$y_r'''(x_r) = \frac{60y_e}{x_e^3} \left[6 \left(\frac{x_r}{x_e} \right)^2 - 6 \left(\frac{x_r}{x_e} \right) + 1 \right]$$

Equação 9 - Terceira derivada da trajetória do modelo, dado um ponto inicial (x_e, y_e)

$$y_r''''(x_r) = \frac{360y_e}{x_e^4} \left[2 \left(\frac{x_r}{x_e} \right) - 1 \right]$$

Equação 10 - Quarta derivada da trajetória do modelo, dado um ponto inicial (x_e, y_e)

As seguintes curvas são geradas a partir dos seguintes (x_e, y_e) iniciais (mostrados na figura):

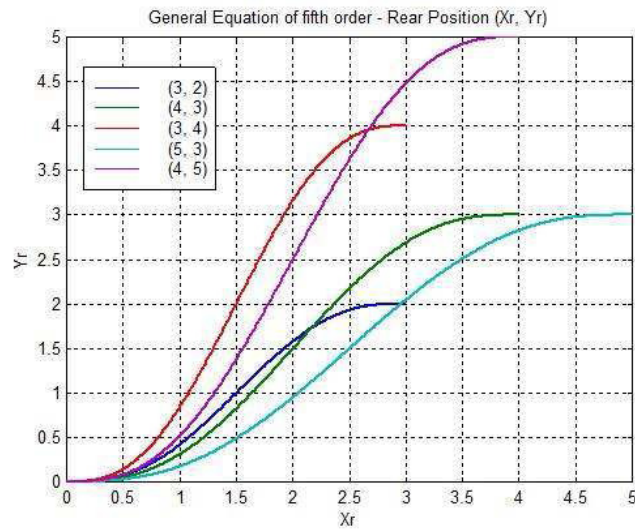


Figura 54 – Equações geradas a partir de diferentes pontos iniciais

Uma vez definida a trajetória, há grande interesse em determinar qual é o tamanho de vaga ideal, dadas as características do carro como comprimento, largura e ângulo máximo de

direção. Para a determinação dessas condições, considere o modelo cinemático do movimento de ré proposto em [1].

$$\dot{x}_r = v \cdot \cos \theta \cos \phi$$

$$\dot{y}_r = v \cdot \sin \theta \cos \phi$$

$$\dot{\theta} = v \cdot \frac{\sin \phi}{l}$$

Equação 11 – Modelo cinemático do movimento para trás.

Começando as contas pela equação da trajetória de referência de quinta ordem, temos que:

$$\frac{d}{dx_r}[y_r] = y'_r = \tan \theta$$

Equação 12 – Derivada primeira da equação de referência é igual à tangente da inclinação do carro com o eixo X

Derivando a equação acima encontramos:

$$y''_r = \sec^2 \theta * \theta'$$

Equação 13 – Derivada da equação 12.

Onde $\sec^2 \theta = 1 + \tan^2 \theta$ e $\theta' = \frac{d}{dx_r}[\theta]$.

Isolando θ' e resolvendo $\sec^2 \theta$ em termos de y'_r , temos:

$$\theta' = \frac{y''_r}{1 + y'^2_r}$$

Equação 14 – Isolando θ' em termos de y_r

Pela derivação parcial temos:

$$\frac{d}{dx_r}[\theta] = \frac{d}{dt}[\theta] * \frac{d}{dx_r}[t] = \dot{\theta} * \frac{1}{\dot{x}_r}$$

Equação 15 – Derivação parcial de θ'

Assim:

$$\frac{y''_r}{1 + y'^2_r} = v \cdot \frac{\sin \phi}{l} * \frac{1}{v \cdot \cos \theta \cos \phi}$$

Isolando em ϕ , temos:

$$\tan \phi = l * \cos \theta * \frac{y''_r}{1 + y'^2_r}$$

Como $\cos \theta = \sqrt{\frac{1}{\sec^2 \theta}} = \frac{1}{\sqrt{1 + y'^2_r}}$, substituindo temos:

$$\tan \phi = l * \frac{y_r'''}{\sqrt{(1 + y_r'^2)^3}} = l * k$$

Equação 16 – Equação que relaciona ϕ à equação da curva de referência

Onde k é a curvatura. Dessa forma chegamos ao mesmo resultado mostrado em [3]. As equações foram mostradas para uma curva de referência $y_r = f(x_r)$ qualquer. Em [3] são mencionadas outras formas de curvas de estacionamento com diferentes características de aproximação.

Usando as informações da função anterior do programa de controle para definir o ambiente, para as dimensões do carro utilizado, $(x_e, y_e) = (750, 250) \text{ mm}$ é um resultado provável. Como o ângulo máximo permitido para o modelo é de 20° , o carro não consegue executar a curva de estacionamento de quinta ordem proposta, uma vez que o ângulo máximo de direção para essa curva solicitado do carro é de $35,63^\circ$, conforme figuras abaixo.

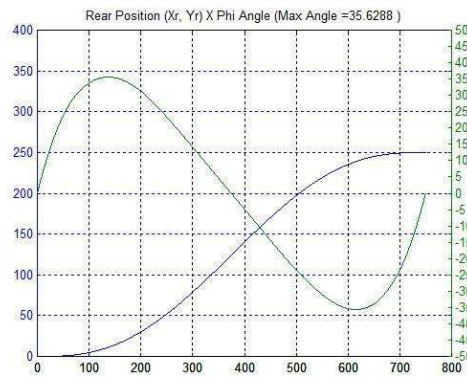


Figura 55 – Posição da eixo traseiro versus evolução do ângulo ϕ para $(x_e, y_e) = (750, 250) \text{ mm}$

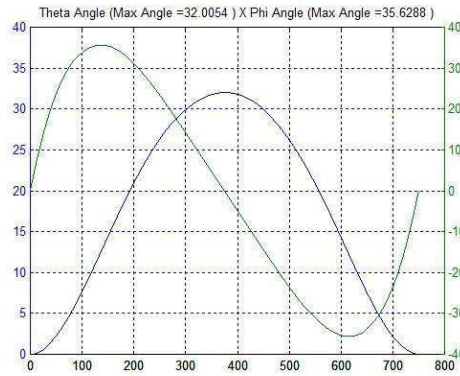


Figura 56 – Evolução do ângulo θ versus ângulo ϕ para $(x_e, y_e) = (750, 250) \text{ mm}$

Como pode ser visto nas figuras acima, quando a concavidade da curva de referência muda (segunda derivada igual a zero), o ângulo ϕ atinge o valor zero.

Variando os parâmetros para encontrar uma curva que caiba na limitação de ângulo de direção de 20° , encontramos o seguinte ponto de início de estacionamento $(x_e, y_e) = (1410, 410) \text{ mm}$ que será usado na próxima função desenvolvida.

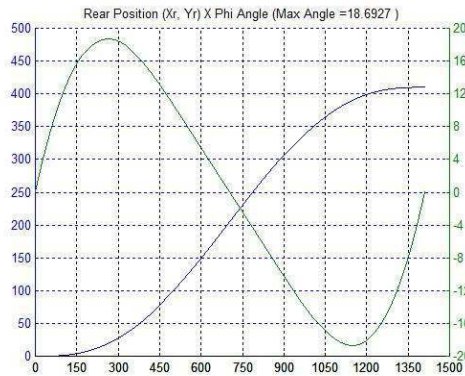


Figura 57 – Posição do eixo traseiro versus evolução do ângulo ϕ para $(x_e, y_e) = (1410, 410) \text{ mm}$

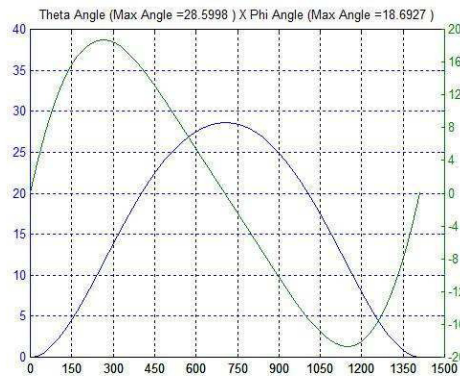


Figura 58 – Evolução do ângulo θ versus ângulo ϕ para $(x_e, y_e) = (1410, 410) \text{ mm}$

1.5 ESTACIONAR CARRO

O algoritmo para o estacionamento do carro é:

Para estacionar o carro, a vaga e o ambiente devem ser definidos e o carro deve ser posicionado. A função “Procurar Vaga de Estacionamento” captura as informações necessárias para a criação do ambiente e posiciona o carro na fronteira da borda.

Pelo controlador utilizado, temos que mirar em algum ponto da trajetória de referência para que o carro realize as manobras de estacionamento. Foram criadas duas heurísticas para selecionar um ponto sobre a trajetória de referência:

- Seguimento por passo constante – o ponto alvo (x, y) escolhido estará sempre a uma distância fixa em $P \text{ mm}$ do centro do eixo de rodas traseiro (x_r, y_r) .
- Seguimento por tempo constante – o ponto alvo (x, y) escolhido será o centro do eixo de rodas traseiro de um carro imaginário igual ao carro modelo e que sempre esteja sobre a curva de referência.

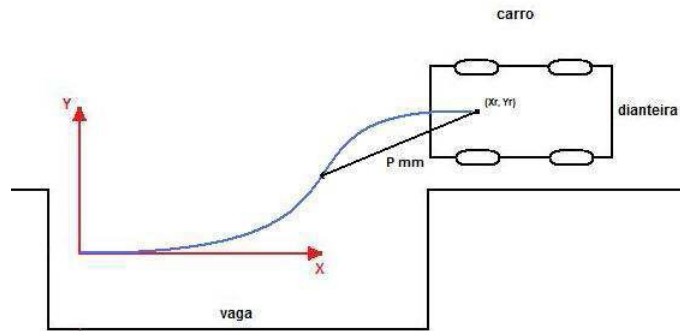


Figura 59 – Heurística “Seguimento de passo constante”

1.5.1 SEGUIMENTO POR PASSO CONSTANTE

A heurística de seguimento por passo constante consiste em:

Dada uma posição do carro, sempre há um vetor P de tamanho constante que liga o centro do eixo traseiro de rodas (x_r, y_r) a um ponto (x, y) sobre a curva de referência. A curva de referência estudada é uma equação de quinta ordem e o cálculo de distância envolve quadrados e raízes, o que levaria a uma análise de uma equação de décima ordem.

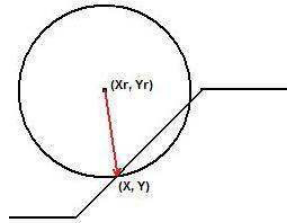


Figura 60 – Aproximação do ponto (x, y) à reta

Aproximando a curva de referência por uma reta com equação $f(x) = ax + b$ e imaginando um círculo de raio P com centro em (x_r, y_r) que toque a reta em pelo menos um ponto, temos:

$$(x - x_r)^2 + (y - y_r)^2 = P^2$$

$$(x - x_r)^2 + (ax + b - y_r)^2 = P^2$$

Equação 17 – Equação da distância do centro do eixo traseiro a uma curva de referência representada por uma reta $ax + b$

Substituindo $b - y_r = c$, temos:

$$x^2 - 2xx_r + x_r^2 + a^2x^2 + 2acx + c^2 = P^2$$

$$(a^2 + 1)x^2 + 2(ac - x_r)x + (x_r^2 + c^2 - P^2) = 0$$

$$x = \frac{-2(ac - x_r) \pm \sqrt{4(ac - x_r)^2 - 4(a^2 + 1)(x_r^2 + c^2 - P^2)}}{2(a^2 + 1)}$$

$$x = \frac{(x_r - ac) \pm \sqrt{(ac - x_r)^2 - (a^2 + 1)(x_r^2 + c^2 - P^2)}}{(a^2 + 1)}$$

Equação 18 – Equação para cálculo de x sobre a curva de referência pela aproximação de 3 segmentos – reta descrita por $ax + b$

Se $a = 0$, ou seja, se a função de referência for uma reta horizontal, a equação fica da seguinte forma:

$$x = x_r \pm \sqrt{P^2 - c^2}$$

Equação 19 – Equação para cálculo de x sobre uma reta horizontal de referência

Como a equação de referência é de quinta ordem, a distância de (x_r, y_r) é aproximada por uma nova curva de referência utilizando três segmentos de reta, como mostrado abaixo.

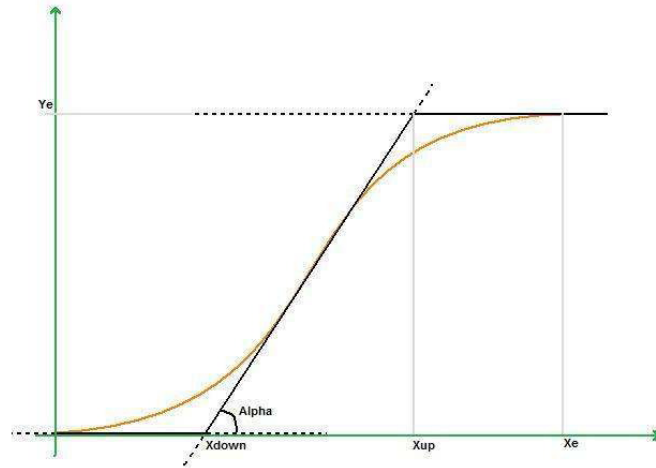


Figura 61 – Seguimento de Passo Constante – 3 segmentos para aproximação

Na aproximação por 3 segmentos, temos que:

$$y_r(x_r) = \begin{cases} 0, & x < x_{DOWN} \\ a x_r + b, & x_{DOWN} \leq x \leq x_{UP} \\ y_e, & x > x_{UP} \end{cases}$$

Onde $a = \tan \alpha$, onde α é o ângulo médio do trecho em que o ângulo do carro com a horizontal é máximo. A ideia por trás da utilização da aproximação por 3 segmentos é que, para diferentes curvas de estacionamento, o comportamento da equação para o trecho central de x_r indica que a função se comporta como uma reta de equação $ax + b$, como pode ser visto na figura 54.

Para calcular os parâmetros a e b , temos:

$$y_r(x_{DOWN}) = 0 = a x_{DOWN} + b$$

$$b = -a x_{DOWN}$$

Equação 20 – Equação para cálculo dos parâmetros a e b - isolando b

$$y_r(x_{UP}) = y_e = a x_{UP} + b = a x_{UP} - a x_{DOWN}$$

$$a = \frac{y_e}{x_{UP} - x_{DOWN}} = \tan \alpha$$

Equação 21 – Equação para cálculo dos parâmetros a e b - isolando a

Para cálculo de x_{DOWN} , x_{UP} e $\tan \alpha$, calculamos a média das inclinações, de y_r e de x_r da curva de referência em torno da região central da curva, conforme mostrado no trecho do programa que executa os cálculos (figura 63). Esses parâmetros são calculados pela função “Definir Trajetória”.

Calculados os parâmetros da função de 3 segmentos de reta, temos duas opções para utilização do x calculado pela aproximação: usar o y da função de 3 segmentos ou calcular o y usando a equação de quinta ordem. O programa de controle mostra as duas opções citadas e mais uma terceira que representa uma curva que conecta diretamente (x_e, y_e) a $(0,0)$, chamada de 1 segmento.

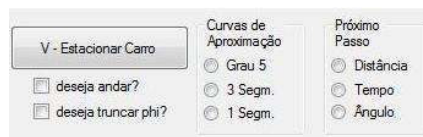


Figura 62 – Parte do programa de controle para a função “Estacionar Carro”

```
trajetoria(double xE, double yE, double pP)
{
    int nSoma = 10 ;

    double yLSoma = 0, ySoma = 0, xSoma = 0 ;

    for (int i=0; i < nSoma ; i++)
        yLSoma = yLSoma + calculaY(1, xE, yE, xE/9.0*(4.0+(double)(i+1)/(double)nSoma)) ;

    yLSoma = (double) yLSoma/(double)nSoma ; // calculo da tangente de thetamedio

    passoP = pP ;
    trajTg = yLSoma ;

    for (int i=0; i < nSoma; i++)
    {
        ySoma = ySoma + calculaY(0, xE, yE, xE/9.0*(4.0+(double)(i+1)/(double)nSoma)) ;
        xSoma = xSoma + xE/9.0*(4.0 + (double)(i+1)/(double)nSoma) ;
    }

    xDown = xSoma/((double)nSoma) - ((ySoma/(double)nSoma)/trajTg) ; // calculo de xDown e xUp
    xUp = xDown + yE / trajTg ; // aproximacao do polinomio de grau 5 por 3 segmentos de retas

    trajXe = xE ;
    trajYe = yE ;
}
```

Figura 63 – Trecho do código que calcula x_{DOWN} , x_{UP} e a tangente do ângulo médio α

A escolha do próximo x segue o seguinte fluxograma:

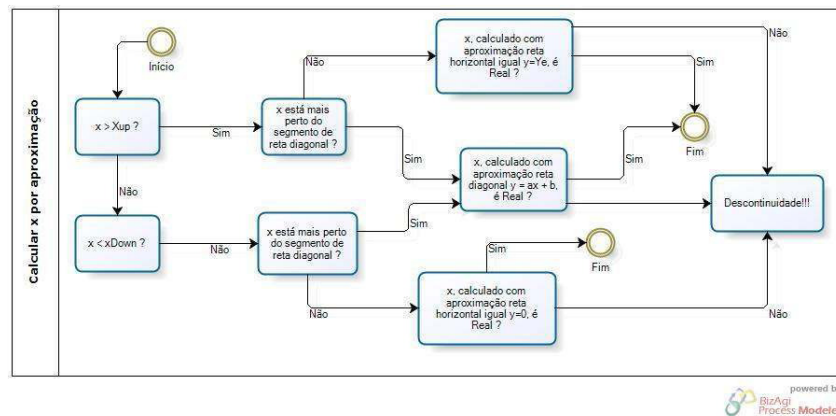


Figura 64 – Fluxograma de cálculo do próximo x para o seguimento por passo constante

No processo acima, verificar se o x é \Re implica em verificar se os termos das raízes descritas nas equações 18 e 19 são positivos. Caso algum x calculado não seja \Re , o círculo de raio P em torno de (x_r, y_r) não toca a curva de referência. A aproximação utilizada para calcular o próximo passo foi decrementar em P vezes o cosseno de 72° , como ângulo θ máximo para o carro.

Como pode ser visto no gráfico abaixo, nos pontos em que o passo P muda do segmento de reta horizontal para o diagonal, na transição em x_{UP} , e muda do segmento de reta diagonal para o horizontal, na transição em x_{DOWN} , há uma pequena descontinuidade na curva de referência, o que faz com que os parâmetros u_1 e u_2 tenham descontinuidades no cálculo de seus valores.

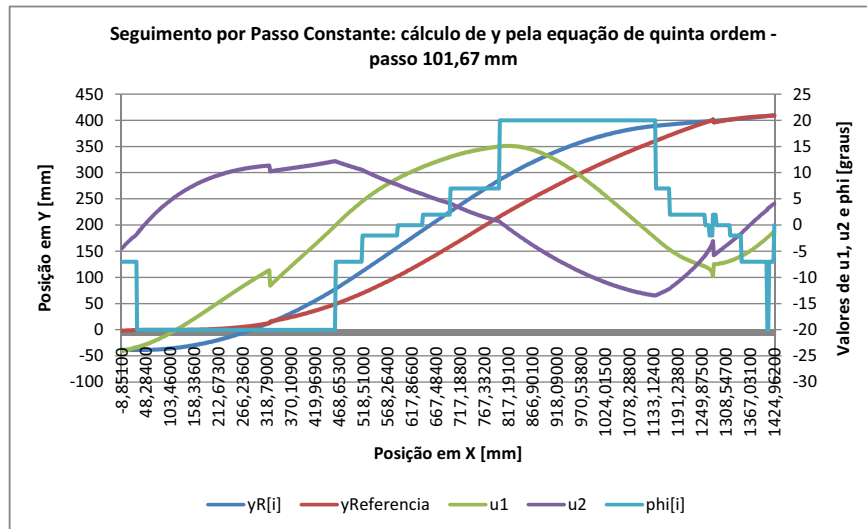


Figura 65 – Gráfico das trajetórias executadas pelo modelo e usadas como referência – passo $P = 101,67$ mm

Com a redução do passo para 50 mm, aumenta-se a probabilidade de que o círculo de raio P não toque a curva de referência e, com isso, aumentam os trechos e os efeitos da descontinuidade. Note que no gráfico 66 a imagem das variáveis u_1 e u_2 é maior. No entanto, esse resultado apresenta uma melhor aproximação, uma vez que o erro final é menor do que 10 mm.

Com a redução do passo para 30 mm, aumentam ainda mais os trechos e os efeitos da descontinuidade. Note que no gráfico 67 a imagem da variável u_1 é maior do que a imagem da curva para o passo de 50 mm. No entanto, esse resultado apresenta uma melhor aproximação, mas uma maior variação no ângulo de direção ϕ na aproximação final.

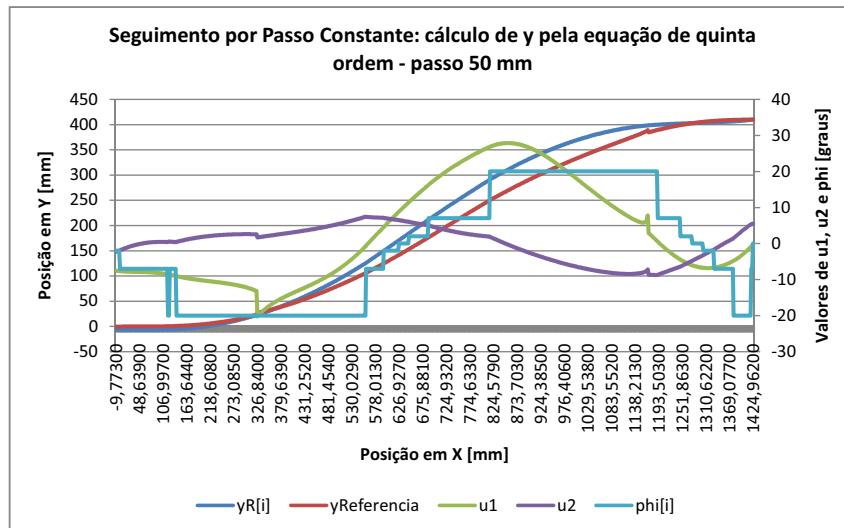


Figura 66 – Gráfico das trajetórias executadas pelo modelo e usadas como referência – passo P = 50 mm

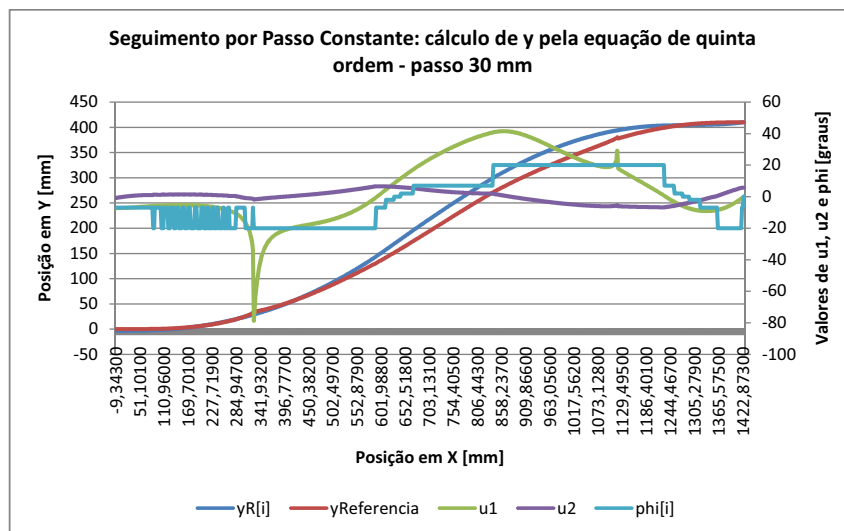


Figura 67 – Gráfico das trajetórias executadas pelo modelo e usadas como referência – passo P = 30 mm

1.5.2 SEGUIMENTO POR TEMPO CONSTANTE

A heurística do seguimento por tempo constante usa os parâmetros ϕ calculado pelo controlador fuzzy e θ calculado pela equação 12 para inferir o valor do próximo x da curva de

referência. Uma vez calculado, o novo x é usado na fórmula de $y_r = f(x_r)$ para calcular y e θ . Resumindo:

$$x_{i+1} = x_i - v * dt * \cos \theta_i * \cos \phi_i$$

$$y_{i+1} = f(x_{i+1})$$

$$\theta_{i+1} = f'(x_{i+1})$$

Equação 22 – Cálculo dos próximos pontos da curva de referência.

$$\phi_{i+1} = \text{controladorFuzzy}(u_1, u_2)$$

Equação 23 – Resultado do controlador fuzzy para as variáveis u_1 e u_2 .

Os parâmetros $v * dt$ representam a distância percorrida pelo modelo da iteração anterior até a aplicação do resultado do controlador da iteração corrente. De acordo com o modelo matemático proposto pela equação 11, esses dois parâmetros são os pontos mais críticos do modelo, uma vez que não há realimentação via câmera e, portanto, (x_r, y_r) não são obtidos de forma absoluta.

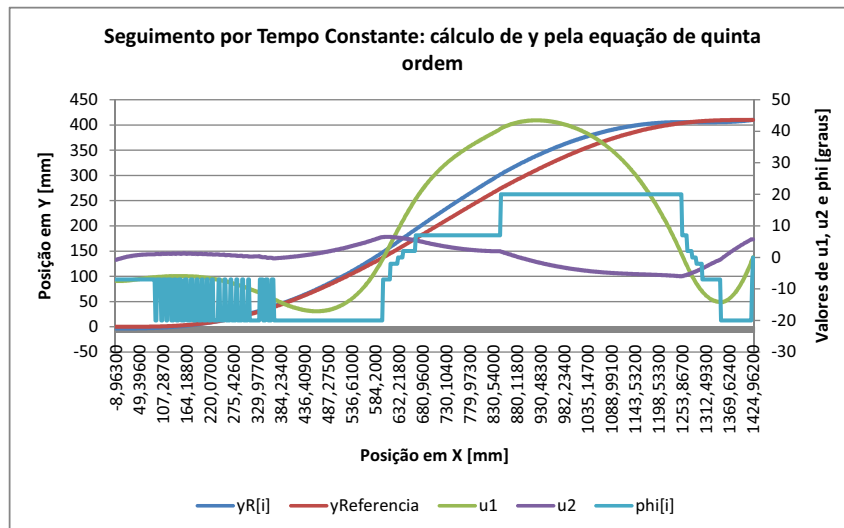


Figura 68 - Gráfico das trajetórias para o seguimento por tempo constante

A solução por tempo constante apresenta a melhor aproximação em relação a posição em y_r final, mas executa mais manobras no ângulo de direção ϕ . A execução de manobras rápidas e repetitivas em ϕ não é desejável pois o controlador ou o modelo podem não responder fisicamente aos comandos enviados.

ANEXO B – Firmware da Placa de Controle

```

/*****
Firmware - Placa de controle
Project : Placa de controle
Version :
Date   : 13/12/2008
Author : TIAGO REINAN
Company : Notebook
Comments:
Chip type      : mega32
Program type   : Application
Clock frequency : 8,000000 MHz
Memory model   : Small
External SRAM size : 0
Data Stack size : 512
*****/

#include <mega32.h>
#include <delay.h>
#include <stdio.h>
#define led PORTB.6
#define cts PINB.4
#define direcao OCR1A
#define centro 1156 //1152    // para direcao
servo1_centro = 1340 +- 300
#define limite_esq 1476//1472//1560
#define limite_dir 836 //832//840
#define motor OCR1B
#define neutra 1500
#define limite_max_motor 1900
#define limite_min_motor 1100
#define buffer_size 750
#define folga 60 //128
#define direito '-'
#define esquerdo '+'

// Declare your global variables here
unsigned int buffer[buffer_size];
char buffer_encoder[buffer_size/2];
unsigned int b,n,aux1,aux2;//,somad_cont_encoder;

char
somad_cont_encoder,somad_cont_encoder_aux,comando
;

char
soma_cont_encoder,espera_cont_encoder,cont_encoder;

unsigned char set_point_velo,multiplo_tempo;

unsigned int z;

unsigned char
sentido,lado,sentido_velocidade='+',parar_movimento;

signed char grau,novo_grau;

signed int erro,kp,term;

//signed int ki,is;

bit flag_enviar=0;

#define ADC_VREF_TYPE 0x00
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);

    // Start the AD conversion
    ADCSRA|=0x40;

    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);

    ADCSRA|=0x10;

    return ADCW;
}

/*****INTERRUPÇÕES*****/

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{

```



```

cont_encoder++;
//soma_cont_encoder++;
somad_cont_encoder++; //
GIFR=0x40;
}
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
//entra a cada 33 ms
{
    if(parar_movimento)
    {
        erro=-(signed int)cont_encoder; //
        set_point_velo=0
    }
    else
        erro=(signed int)set_point_velo-(signed
int)cont_encoder;
    kp=erro/7;
    if(erro%7>3)
        kp++;
    is+=erro;
    ki=is/900;
    if(ki>70)
        ki=70;
    if(ki<-70)
        ki=-70;
    term=ki+kp;
    if(parar_movimento)
    {
        if(erro==0)
        {
            parar_movimento=0;
            if(sentido_velocidade=='+')
                sentido_velocidade='-';
            else sentido_velocidade='+';
        }
    }
}
if(sentido_velocidade=='-') //
{
    if(cont_encoder==0)
    {
        espera_cont_encoder++;
        if(espera_cont_encoder==80)
        {
            motor=neutra;
            espera_cont_encoder=0;
        }
    }
}
if(sentido_velocidade=='+') // frente
{
    if(motor+term>neutra+70)
    {
        term=neutra+70;
        motor=term;
    }
    else
    {
        if(term<0)
        {
            if(motor+term<neutra)
            {
                term=neutra;
                motor=term;
            }
        }
        else
    }
}

```

```

        motor+=term;
    }
else
    {
        motor+=term;
    }
}

} // if sentido_velocidade
} // else

} // if sentido_velocidade
else
{
    if(sentido_velocidade=='-') // re
    {

        if(motor-term<neutra-70)
        {
            term=neutra-70;
            motor=term;
        }
        else
        {
            if(term<0)
            {
                if(motor-term>neutra)
                {
                    term=neutra;
                    motor=term;
                }
                else
                {
                    motor+=-term;
                }
            }
            else
            {
                {
                    motor+=-term;
                }
            }
        }
    }

    // Timer 2 overflow interrupt service routine
    interrupt [TIM2_OVF] void timer2_ovf_isr(void)
    {
        if (n==6)// MULTIPLO DE 6*4.096 MS
        {
            buffer_encoder[b/2]=soma_cont_encoder;
            buffer[b++]=read_adc(0);
            n=0;
            buffer[b++]=read_adc(1);

            if (b==buffer_size)
            {
                motor=neutra;
                set_point_velo=0;
                //TCCR0=0x00;
                led=1;
            }
        }
    }
}

```

```

        TCCR2=0x00;

    }

}

n++;

}

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
    signed int aux;
    unsigned int i;
    switch (getchar())
    {

case 'a': //A
    if (direcao<limite_esq+folga)
    {
        if (sentido==direito)
        {

            direcao+=16+folga;

        }
        else
        direcao+=16; //16 EQUIVALE A UM GRAU
        grau++;
        sentido=esquerdo;

    }

    //printf("E   direcao:%u   \r",direcao);

break;

case 'd'://D
    if (direcao>limite_dir)
    {
        if (sentido==esquerdo)
        {

            direcao=direcao-16-folga;

        }
        else
        direcao=direcao-16; //16 EQUIVALE A UM
GRAU
        grau--;
        sentido=direito;

    }
    //printf("D   direcao:%u   \r",direcao);

break;

case 'v':
lado=getchar();
novo_grau=getchar()-48;
novo_grau=novo_grau*10+(getchar()-48);
//printf("novo_grau:%c%c%u\r\n",lado,novo_grau);

if(novo_grau>20) novo_grau=20;
if(novo_grau<-20) novo_grau=-20;
if((novo_grau>=0)&&(novo_grau<=20))
{

    if (lado==direito) //D NEGATIVO MENORES
VALORES

```

```

    {
        novo_grau=novo_grau*-1;
    }
    aux=novo_grau-grau;
    if (aux>0) //VOU PARA O SENTIDO PARA
    ESQUERDA
    {
        if ( sentido==direito) // ESTAVA INDO PARA
    DIREITA
        {
            direcao=direcao+16*aux+folga;
        }
        else
        {
            direcao=direcao+16*aux;
        }
        grau=novo_grau;
        sentido=esquerdo;
        // printf("E   direcao:%u
    grau:%d\r\n",direcao,grau);
    }
    else
    {
        if (aux<0) // VOU PARA O SENTIDO
    DIREITA
        {
            if ( sentido==esquerdo) // ESTAVA
    INDO PARA DIREITA
            {
                direcao=direcao+16*aux-folga;
            }
            else
            {
                direcao=direcao+16*aux;
            }
            grau=novo_grau;
            sentido=direito;
            //printf("D   direcao:%u
            grau:%d\r\n",direcao,grau);
        }
    }
    //send_ok(data_array[6]); // ok da sequencia recebida
    }
    break;

    case 't':// usado para aplicacoes especificas devolve alfa 3
    e alfa2
    aux1=read_adc(0); // pego dados no inicio para evitar o
    atra so na leitura
    aux2=read_adc(1);
    somad_cont_encoder_aux=somad_cont_encoder;
    lado=getchar();
    novo_grau=getchar()-48;
    novo_grau=novo_grau*10+(getchar()-48);
    //printf("novo_grau:%c%u\r\n",lado,novo_grau);
    if(novo_grau>20) novo_grau=20;
    if(novo_grau<-20) novo_grau=-20;
    if((novo_grau>=0)&&(novo_grau<=20))

```

```

{
    if (lado==direito) //D NEGATIVO MENORES
    VALORES
    {
        novo_grau=novo_grau*-1;
    }
    aux=novo_grau-grau;
    if (aux>0) //VOU PARA O SENTIDO PARA
    ESQUERDA
    {
        if ( sentido==direito) // ESTAVA INDO PARA
        DIREITA
        {
            direcao=direcao+16*aux+folga;
        }
        else
        {
            direcao=direcao+16*aux;
        }
        grau=novo_grau;
        sentido=esquerdo;
        // printf("E   direcao:%u
        grau:%d\r\n",direcao,grau);
    }
    else
    {
        if (aux<0) // VOU PARA O SENTIDO
        DIREITA

```

```

{
    if ( sentido==esquerdo) // ESTAVA
    INDO PARA DIREITA
    {
        direcao=direcao+16*aux-folga;
    }
    else
    {
        direcao=direcao+16*aux;
    }
    grau=novo_grau;
    sentido=direito;
    //printf("D   direcao:%u
    grau:%d\r\n",direcao,grau);
}
}

//aux1=0;
//aux2=0;
//for (i=0;i<16;i++)
//{

//aux1=read_adc(0);
//aux2=read_adc(1);
//}

//aux1/=16;
//aux2/=16;

```

```

//somad_cont_encoder_aux=somad_cont_encoder;          direcao=direcao+16*aux+folga;
//while(cts==1);                                     }
//printf("%u %u %u\n",aux1,aux2,somad_cont_encoder);    else
flag_enviar=1;                                         {
comando='t';                                           direcao=direcao+16*aux;
}                                                       }
break;                                                 grau=novo_grau;
                                                       sentido=esquerdo;
                                                       // printf("E   direcao:%u
                                                       grau:%d\n",direcao,grau);

case 'p':// manda alfa1 do pot
                                                       }
lado=getchar();
novo_grau=getchar()-48;
novo_grau=novo_grau*10+(getchar()-48);
                                                       else
//printf("novo_grau:%c%u\n",lado,novo_grau);          {
if(novo_grau>20) novo_grau=20;
if(novo_grau<-20) novo_grau=-20;
if((novo_grau>=0)&&(novo_grau<=20))
{
                                                       if (aux<0) // VOU PARA O SENTIDO
DIREITA
{
                                                       if ( sentido==esquerdo) // ESTAVA
INDO PARA DIREITA
{
                                                       direcao=direcao+16*aux-folga;
}
                                                       else
{
                                                       direcao=direcao+16*aux;
}
                                                       grau=novo_grau;
                                                       sentido=direito;
                                                       //printf("D   direcao:%u
                                                       grau:%d\n",direcao,grau);
if (lado==direito) //D NEGATIVO MENORES
VALORES
{
                                                       direcao=direcao+16*aux-folga;
}
                                                       else
{
                                                       direcao=direcao+16*aux;
}
                                                       grau=novo_grau;
                                                       sentido=direito;
                                                       //printf("D   direcao:%u
                                                       grau:%d\n",direcao,grau);
if (aux>0) //VOU PARA O SENTIDO PARA
ESQUERDA
{
                                                       if ( sentido==direito) // ESTAVA INDO PARA
DIREITA
{

```

```

    }
}

if(z<buffer_size)    // salva os angulos em buffer[]
{
    buffer[z++]=read_adc(0);

    buffer[z++]=read_adc(1);
}

if (z>=buffer_size)
{
    motor=neutra;
    set_point_velo=0;
    TCCR0=0x00;
    led=1;
}

}

break;

case 'w'://W
TCCR0=0x00;
set_point_velo=0;
    if (motor<limite_max_motor) motor+=10;
    // printf("+   motor:%u   \r",motor);

break;

case 's'://S
TCCR0=0x00;

set_point_velo=0;
    if (motor>limite_min_motor) motor-=10;
    // printf("-   motor:%u   \r",motor);

break;

case 'n'://S
TCCR0=0x00;

motor=neutra;
set_point_velo=0;

break;

case 'o': // seta a velocidade do carro
if(getchar()!=sentido_velocidade) // usado quanto freiar o
truck antes da inversao do sentido da velocidade
{
    parar_movimento=1;
    //sentido_velocidade_aux=rx_buffer[1];
}

aux=getchar()-48;
if(aux<10)
{
    cont_encoder=0;
    set_point_velo=2*aux;
    //set_point_velo=set_point_velo*2;
    TCNT0=0x00;
    TCCR0=0x05;

}

break;

```

```

case 'i':
    multiplo_tempo=getchar()-48;
    //multiplo_tempo=10*multiplo_tempo+rx_buffer[2]-48;
    TCCR2=0x05;
    b=0;
    n=0;
    sentido_velocidade='+';
    set_point_velo=8;
    TCNT0=0x00;
    TCCR0=0x05;
    cont_encoder=0;
    break;

case 'c':
    motor=neutra;
    WDTCR=0x08;
    while(1);
    break;

case 'r':
    i=0;
    while(i<buffer_size) //
    {

        while(cts==1);

        printf("%u\t%u\t%u\t%u\t%u\r\n",i/2+1,buffer[i],buffer[i+1],buffer_encoder[i/2]);
        i+=2;
        //delay_ms(30);
    }
    break;

case 'z':
    aux1=read_adc(0);
    aux2=read_adc(1);
    //delay_ms(1000);
    flag_enviar=1;
    comando='z';
    break;

case 'k': // le sensores laser, douglas
    aux1=read_adc(0);
    aux2=read_adc(1);
    //aux3=read_adc(5);
    //aux4=read_adc(5);
    //while(cts==1);
    //printf("A %uB %u",aux1,aux2);//,aux3,aux4);
    break;

case 'm':// manda valor da ultima contagem do encoder
    //i=somad_cont_encoder;
    somad_cont_encoder_aux=somad_cont_encoder;
    flag_enviar=1;
    comando='m';
    break;

}

//UCSRB=0x98;

}

/** FUNCOES***/

void pisca_led(char n)
{

```



```
char i;

for (i=0;i<n;i++)
{
    led=1;
    delay_ms(500);
    led=0;
    delay_ms(500);
}

}

void calibra_esc(void)
{
    led=1;
    delay_ms(500);
    led=0;
    delay_ms(500);
    led=1;

    motor= neutra;
    delay_ms(7000);

    led=0;
    delay_ms(500);
    led=1;
    motor=limite_max_motor;
    delay_ms(7000);
    led=0;
    delay_ms(500);
    led=1;
    motor=limite_min_motor;
    delay_ms(7000);
    led=0;
}

void init_direcao(void){
    direcao=limite_esq;
    delay_ms(500);
    direcao-=folga;
    delay_ms(500);
    direcao=centro;
    sentido=direito;
    grau=0;
}

void main(void)
{
    // Declare your local variables here
    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In
    Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T
    State2=T State1=T State0=T
    PORTA=0x00;
    DDRA=0x00;

    // Port B initialization
    // Func7=In Func6=Out Func5=In Func4=In Func3=In
    Func2=In Func1=In Func0=In
    // State7=T State6=0 State5=T State4=T State3=T
    State2=T State1=T State0=T
    PORTB=0x00;
    DDRB=0x40;

    // Port C initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In
    Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T
    State2=T State1=T State0=T
```

```

PORTC=0x00;

DDRC=0x00;

// Port D initialization

// Func7=Out Func6=Out Func5=Out Func4=Out
Func3=In Func2=In Func1=In Func0=In

// State7=0 State6=0 State5=0 State4=0 State3=T
State2=P State1=T State0=T

PORTD=0x00;

DDRD=0xF0;

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: 7,813 kHz

// Mode: Normal top=FFh

// OC0 output: Disconnected

TCCR0=0x00;

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: 1000,000 kHz

// Mode: Fast PWM top=ICR1

// OC1A output: non Inverted

// OC1B output: Inverted

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer 1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

//TCCR1A=0xF2;

TCCR1A=0xB2; //no invert OC1A

TCCR1B=0x1A;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x4E; // =19999 => pwm com periodo de
20ms

ICR1L=0x1F;

motor=neutra;//neutra

direcao=centro;//centro;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: 62,500 kHz

// Mode: Normal top=FFh

// OC2 output: Disconnected

ASSR=0x00;

TCCR2=0x00;

TCNT2=0x00;

OCR2=0x00;

// External Interrupt(s) initialization

// INT0: On

// INT0 Mode: Any change

// INT1: Off

// INT2: Off

GICR=0x40;

MCUCR=0x01;

MCUCSR=0x00;

GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=0x41;

// USART initialization

```

```

// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 38400
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x0C;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1:
Off
ACSR=0x80;
SFIOR=0x00;
// ADC initialization
// ADC Clock frequency: 250,000 kHz
// ADC Voltage Reference: AREF pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x85;
#asm("sei")
//printf("ola\r\n");
//calibra_esc();
led=1;
delay_ms(500);
led=0;
delay_ms(500);

led=1;
delay_ms(500);
led=0;
init_direcao();

//printf("valendo\r\n");
//TCCR0=0x05;
//motor+=20;
//delay_ms(3000);
TCCR0=0x05;
while (1)
{
    if( flag_enviar)
    {
        switch(comando)
        {
            case 't':
                //while(cts==1);
                printf("%u %u\n",aux1,aux2);
                break;

            case 'z':
                // while(cts==1);
                printf("%u\t%u\r\n",aux1,aux2);
                break;

            case 'm':
                //while(cts==1);
                printf("%u\n",somad_cont_encoder_aux);
                break;
        }
        flag_enviar=0;
    }
}

```

ANEXO C – Algoritmo de Controle

```
% PROGRAMA DE CONTORLE DE CONFIGURAÇÃO - C2t_1.m
%MATLAB 7.1
%TRUCK-TRAILER- TRAILER

LB0 = 0;
a1 = 20;      % angulo de direcao maximo
LA1 = 30;     % comprimento do eixo do trator as rodas dianteiras
LB1 = 0;
LA2 = 49.5;   % comprimento do eixo do trailer 1 as rodas dianteiras
LB2 = 26.5;   % comprimento do eixo do trailer 1 ao engate
LA3 = 49.5;   % comprimento do eixo do trailer 2 as rodas dianteiras

warning('on');
pot_ang=2.844;      %fator de conversao de AD para angulos
a2_pot_zero=319;    % valor a2 alinhado
a3_pot_zero=432;    % valor a3 alinhado
m=[0 0 0];
x=[];
i=0;
controlador1=readfis('contag_1_1_erro'); % carrega fuzzy
controlador2=readfis('contag_2_2_erro_pos');
a3_ref=60; % referencia para a3
a3=0;
a2=0;
a1=0;
i=0;
k=0;
soma1=0;
soma3=0;
kd=0;
erro3_ant=0;
vet_erro3=[0 0 0 0 0 0 0 0 0];
vet_erro=[0 0 0 0 0 0 0 0 0];
encoder_aux=0;
distancia=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SERIAL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s1=serial('COM6', 'BaudRate',38400,'timeout',0.1);
get(s1,{'BaudRate','DataBits','Parity','StopBits','FlowControl','Terminator'})
fopen(s1);
pause(1);
fwrite(s1,'o');
fwrite(s1,'+');
fwrite(s1,'2'); % seta velocidade
fprintf(s1,'t%+03d',round(a1)); %
pause(0.01);
m= fscanf(s1,'%d%d%d');
a2=m(1,1); %
a3=m(2,1); %
a2=(a2-a2_pot_zero)/pot_ang % mostra a2 inicial
a3=(a3-a3_pot_zero)/pot_ang % mostra a3 inicial
fprintf(s1,'t%+03d',round(a1)); % manda a1 e prepara para fscanf do loop

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while (k<450) && (abs(a2)<45) && (abs(a3)<70)) % Loop principal
%while(k<500) %critico 68

m= fscanf(s1,'%d%d%d'); % recebe o valor de a2 e a3 e econdere.
fprintf(s1,'t%+03d',round(a1)); % Envia a1
%pause(0.01);
s=size(m);

if s(1,1)== 3 % checa se o pacote recebito possui tres e elementos
encoder=m(3,1); % pega o contador do encoder
a2_aux=m(1,1); %
a3_aux=m(2,1); %
a2_aux=(a2_aux-a2_pot_zero)/pot_ang; % calcula a2
```

```

a3_aux=(a3_aux-a3_pot_zero)/pot_ang;    % calcula a3

if(encoder<encoder_aux)                  % acrescentar 256 quando o encoder estoura
    distancia_aux=distancia+ encoder+256-encoder_aux;
else distancia =distancia+ encoder-encoder_aux;
end
encoder_aux=encoder;

if (abs(a2_aux-a2)<10&&abs(a3_aux-a3)<10) % NAO PODE PARTIR COM DIFERENCA MAIOR QUE 10 GRAUS DA
INICIALIZACAO DE A2 E A3!!
a2=a2_aux;                               %USADO PARA NAO PERMITIR LEITURA DE UMA VARIACAO BRUSCA,
OU SEJA, ERRO DE RECEPCAO
a3=a3_aux;

erro3_atual=a3_ref-a3;

a2g=atand(LA2*abs(sind(a3))/(LA3+LB2*cosd(a3))); % calcula a2 de giro
if(a3<0)
a2g=-a2g; % ajusta sinal
end

a1g=atand(LA1*abs(sind(a2))/LA2); % calcula a1 de giro
if(a2>0)
a1g=-a1g; %inverte o sinal de a1g, devido a referencias
end
for j=1:9 % janela do erro3
    vet_erro3(j)=vet_erro3(j+1);
end
vet_erro3(10)=erro3_atual;
soma13=0;
for j=1:10
    soma13=vet_erro3(j)+soma13;
end

if (soma13 >80) % satura a soma
    soma13=80;
else if (soma13<=-80)
    soma13=-80;
end
end

kd=(erro3_atual-erro3_ant);
if (kd >2)
    kd=2;
else if (kd<-.2)
    kd=-.2;
end
end
erro3_ant=erro3_atual;
erro3=erro3_atual+soma13/40 +kd/2 ; %sinal de erro a3
if (erro3 > 70)
    erro3=70;

else if (erro3<-.70)
    erro3=-.70;
end
end

delta2 = evalfis([erro3 a2],controlador2); % FUZZY
a2_ref=a2g+delta2;
if(a2_ref > 35)
    a2_ref=35;
else if (a2_ref<-35)

```

```

        a2_ref=-35;
    end
end

erro2_atual=a2_ref-a2;

for j=1:9 % janela do erro2
    vet_erro(j)=vet_erro(j+1);
end
vet_erro(10)=erro2_atual;
somai=0;

for j=1:10
    somai=veter_erro(j)+somai;
end

if (somai >160)
    somai=160;
    if (somai<-160)
        somai=-160;
    end
end

erro2=erro2_atual+somai/10; %sinal de erro a2

if(erro2>70)
    erro2=70;
else if (erro2<-70)
    erro2=-70;
end
end

delta1 = evalfis([erro2],controlador1); % FUZZY
a1=a1g+delta1;
if(a1 > 20)
    a1=20;
else if (a1<-20)
    a1=-20;
end
end

x = [x; distancia a1 a2 a2_ref delta1 a3 a3_ref erro3_atual delta2]; % salva dados
k=k+1;
end

else pause(0.06);
end %end do if s(1,1)

end %end do while

fwrite(s1,'n'); % para o caminhao
fclose(s1); % fecha a serial
save('teste1','x');
plot(x(:,1),x(:,2),'b',x(:,1),x(:,3),'r',
x(:,1),x(:,4),'k',x(:,1),x(:,5),'m',x(:,1),x(:,6),'g',x(:,1),x(:,7),'k',x(:,1),x(:,8),'c',x(:,1),x(:,
9),'y');
xlabel('Distancia (cm)');
legend('a1','a2','a2_ref','delta1','a3','a3_ref','erro3_atual','delta2');

```